National Library of Canada
Acquisitions and Bibliographic Services Branch
395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada
Direction des acquisitions et des services bibliographiques
395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file *Votre référence*

Our file *Notre référence*

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

UNIVERSITY OF ALBERTA

THE EFFECT OF ALONGSHORE TOPOGRAPHIC VARIATION AND BOTTOM FRICTION ON SHELF WAVE INTERACTIONS

BY

FRANÇOIS W. PRIMEAU

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE

IN

APPLIED MATHEMATICS

DEPARTMENT OF MATHEMATICS

EDMONTON, ALBERTA

FALL, 1992

National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-77314-6

Canada

UNIVERSITY OF ALBERTA

RELEASE FORM

Name of Author	François W. Primeau
TITLE OF THESIS	The effect of alongshore topographic variation and bottom friction on shelf wave interactions
Degree for which Thesis was Presented	Master of Science
Year of Degree Granted	Fall 1992

Permission is hereby granted to the UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extracts from it may be printed or otherwise reproduced without the author's written permission.

(Signed) François Primeau

Permanent Address:

898 Miller Street
Sudbury, Ontario, Canada
P3A 2X1

Dated July 31 1992

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a Thesis entitled THE EFFECT OF ALONGSHORE TOPOGRAPHIC VARIATION AND BOTTOM FRICTION ON SHELF WAVE INTERACTIONS submitted by François W. Primeau in partial fulfillment for the requirements for the degree of Master of Science in Mathematics.

(supervisor) G. Swaters

H.I. Freedman

T. Spanos

T.B. Moodie

July 15/92

date

Abstract

The theory of resonant interactions between shelf waves developed by Hsieh and Mysak (1980) is extended to include the effect of bottom friction and alongshore topographic variation. The model equations are derived via a multiple-scale asymptotic expansion in which it is assumed that the alongshore topography varies over a length-scale over which the nonlinear interactions make an order-one contribution to the dynamics. For the case with no bottom friction, it is shown that the total energy of the three interacting waves is conserved and that the Manley-Rowe relations are preserved. Moreover, for the case with no bottom friction and the additional condition that the alongshore topography is restricted to a piecewise constant configuration, the steady state equations can be solved exactly. An illustrative solution for this case is given. It is shown that, in the presence of alongshore topographic forcing, the nonlinear energy exchange will induce a permanent wavenumber mismatch, so that the wave resonance conditions are no longer completely satisfied. For the case with bottom friction, the damping coefficients of each interacting wave is inversely proportional to the phase speed, which in turn depends on the cross shelf profile, but not on the slowly varying alongshore topographic variations. Because the three damping coefficients are not equal the model equations can not be solved exactly. Numerical solutions for some initial conditions are obtained for the purely temporal problem. It is found that the damping due to bottom friction does not induce any wavenumber mismatch.

ACKNOWLEDGEMENTS

I wish to thank Dr. G.E. Swaters for initiating this project and for his helpful scientific advice. I would also like to thank C. Fischer for her technical assistance in preparing this document.

Special thanks goes to a friend C. Kereliuk for making me feel welcome and at home in Alberta.

TABLE OF CONTENTS

Chapter		PAGE
1.	Introduction	1
2.	Governing Equations	5
	2.1 Derivation of the Governing Equations	5
	2.2 Derivation of the Amplitude Equations	10
3.	Conservation Theorems	15
	3.1 Conservation of Energy	15
	3.2 The Manley-Rowe Relations	21
4.	The Theory Applied to an Exponential Shelf Profile	23
	4.1 Resonant Interactions on a Shelf with an Exponential Cross Shelf Profile	23
	4.2 Phase Velocities, Group Velocities, Interaction Coefficients, Friction Coefficients and Topography Coefficients for an Exponential Shelf	26
5.	The Purely Temporal Problem	33
6.	The Steady State Problem	40
	6.1 Steady State Solution With Alongshore Topographic Variation	40
	6.2 Triad Interactions Over a Simple Piece-wise Constant Topographic Configuration	48
7.	Conclusions	73
	Bibliography	76
	Appendix 1: The Group Velocity	76
	Appendix 2: C-Code to Plot Dispersion Curves	79
	Appendix 3: C-Code to Compute a Resonant Triad	82
	Appendix 4: C-Code to Compute Friction Solutions	86
	Appendix 5: C-Code to Compute Steady State Solution	90

LIST OF TABLES

PAGE

Table 1. Coefficients: $b = 1.65$, $H_2 = 14.2$, $r = 2.8$, $h_1 = h_1(Y)$ 30

Table 2. Coefficients: $b = 1.65$, $H_2 = 14.2$, $r = 0.0028$, $h_1 = h_1(Y)$ 30

Table 3. Coefficients: $b = 1.65$, $H_2 = 14.2$, $r = 2.8$, $h_1 = h_1(Y)$ 31

LIST OF FIGURES

PAGE

Figure 1. Dispersion Relation 25

Figure 2. Parallelogram for a Resonant Triad 27

Figure 3. A_j versus time (r = 25.0) 36

Figure 4. A_j versus time (r = 0.8) 37

Figure 5. Alongshore Topography Profile 49

Figure 6. b_j versus Y for initial amplitudes (6.2.10), initial phases (6.2.9) and topography given by (6.2.1) with H = 0.0 56

Figure 7. Plots for initial amplitudes (6.2.10), intial phases (6.2.9) and topography given by (6.2.1) with H - 0.5, Y_0 = -1.0, and Y_1 = -3.0.

(A) b_j versus Y 57
(B) Φ versus Y 58
(C) $\Phi_j - \mu_{j_0} h_1(Y) / c_{g_j}$ versus Y 59
(D) cos(Φ) versus Y 60

Figure 8. Plots for initial amplitudes (6.2.10), intial phases (6.2.9) and topography given by (6.2.1) with H - 0.5, Y_0 = -1.0, and Y_1 = -2.72.

(A) b_j versus Y 61
(B) Φ versus Y 62
(C) $\Phi_j - \mu_{j_0} h_1(Y) / c_{g_j}$ versus Y 63
(D) cos(Φ) versus Y 64

Figure 9. Plots for initial amplitudes (6.2.10), intial phases (6.2.9) and topography given by (6.2.1) with H - 0.75, Y_0 = -1.0, and Y_1 = -3.0.

(A) b_j versus Y 65
(B) Φ versus Y 66
(C) $\Phi_j - \mu_{j_0} h_1(Y) / c_{g_j}$ versus Y 67
(D) cos(Φ) versus Y 68

Figure 8. Plots for initial amplitudes (6.2.10), intial phases (6.2.9) and topography given by (6.2.1) with H - 0.75, $Y_0 = -1.0$, and $Y_1 = -2.75$.

(A) b_j versus Y 69
(B) Φ versus Y 70
(C) $\Phi_j - \mu_{j_0} h_1(Y) / c_{g_j}$ versus Y 71
(D) $\cos(\Phi)$ versus Y 72

Chapter 1

Introduction

Continental shelves form regions of relatively shallow water adjacent to the oceans' shorelines. Their width varies with geographical location but typically extends to offshore distances the order of 50-150 km. The sea floor slopes gently across the continental shelf from the coast to water depths of about 200 m before dropping rapidly to the abyssal plain.

Wave energy can propagate along the coast. This fact may seem surprising to the uninitiated who are accustomed to seeing surface waves incident on a beach. Nonetheless, coastal trapped waves have been detected through observations, by several coastal stations, over long periods of time. The sloping topography of the continental margin provides an efficient wave guide for the propagation of sea level and current fluctuations over vast distances. These topographically guided oscillations are known as continental shelf waves. They were first observed along the east Australian coast by Hamon (1962), and have since been observed along nearly all the major continental boundaries.

Continental shelf waves typically have amplitudes of a few centimeters, periods of several days, and wavelengths of a few thousand kilometers. Further, they propagate their phase with the coast to the right (left) in the northern (southern) hemisphere.

Continental shelf waves appear to play an important role in the dynamics of Gulf Stream meanders, coastal upwelling, shelf circulation, storm surges, and maritime weather and climate. Moreover, the fluid motion over the shelf is important in the sediment transport, pollutant dispersion, and biological phenomena that occur there. In light of the fact that most of the biological primary productivity of the world's oceans takes place in the relatively fertile surface waters over the continental shelves, the study of the dynamics of continental shelf waves seems worthwhile.

The trapping mechanism for shelf waves is based on the fact that these waves conserve potential vorticity, defined as $(\zeta + f)/h$, where ζ is the vertical component of the relative vorticity, f is the Coriolis parameter, and h is the total fluid depth (Longuet-Higgins 1972). Their associated particle motions consist of large vortices (eddies) of alternating spin that drift along the shelf slope region. If a vortex tube is pushed offshore, h increases so that the absolute (and hence relative) vorticity must increase to keep the potential vorticity constant. The increase in the relative vorticity associated with the displaced vortex tube acts to advect adjacent vortex tubes on and offshore which in turn act to advect the original vortex tube back onshore. In this fashion shelf waves undulate onshore-offshore under the action of the restoring mechanism provided by the shelf slope.

Research in shelf wave dynamics has been active in both the observational and theoretical fronts. Mysak (1980a, 1980b) reviewed advances in shelf wave dynamics on the observational and theoretical fronts. Allen (1980) and Brink (1991) reviewed models of wind-driven currents and coastal trapped waves on the continental shelf.

It is generally agreed that the generation of shelf waves is caused primarily by forcing due to surface wind stress. Other mechanisms can also generate shelf waves. In particular, wave-wave interactions can transfer energy between different wave components. Motivated by the observation of three strong peaks in the coherence spectra of sea levels on the Oregon shelf made by Cutchin and Smith (1973), Hsieh and Mysak (1980) proposed a theory of resonant wave interaction to explain the observed signal. The theory suggests that a first wave, excited by a large scale atmospheric disturbance, transfers energy to two other waves via resonant wave-wave interactions.

While much of the research on the theory of shelf waves has been done using the long-wave equations in which the nonlinear terms have been omitted, nonlinear terms in the governing equations can lead to wave-wave interactions. The

energy exchanges due to nonlinear terms is generally of the order of ε, the Rossby number, which is usually small. However, Hsieh and Mysak showed that when certain resonance conditions are satisfied, the nonlinear terms can produce an important contribution to the dynamics. It was previously believed (Buchwald and Adams 1968) that peaks in the shelf wave spectrum might be expected at frequencies where the dispersion curves have zero slope. The theory being that the energy could not propagate away at frequencies where the group velocity was zero. This explanantion however, was found to be lacking, if it was to be used to explain the observed peaks of Cutchin and Smith (1973). The theory Hsieh and Mysak derived agrees well with the observed signal. Nonetheless, in their concluding remarks, Hsieh and Mysak suggested that bottom friction and alongshore topographic variation (both neglected in their theory) may have important effects on the resonant wave-wave interactions.

In order to test whether wave-wave interaction is a viable mechanism for energy exchange in the presence of bottom friction and alongshore topographic variation, we will, in this thesis, extend the theory of Hsieh and Mysak to include the above mentioned effects. An analogous problem for Rossby waves in the atmospheric context was studied by Cree and Swaters (1991).

The outline of this thesis is as follows: In Chapter 2 we derive the nonlinear, rigid lid and barotropic vorticity equation on an f-plane including Rayleigh bottom friction and topography. We then proceed to develop the equations which will describe the evolution of wave amplitudes via a multiple scale asymptotic expansion in which it is assumed that the alongshore topography varies over a length scale over which the nonlinear interactions make an order-one contribution to the dynamics. Finally the amplitude equations are converted to standard form.

In Chapter 3, we show that in the absence of friction, the equations governing the evolution of the amplitudes conserve energy and preserve the Manley-Rowe

relations. In Chapter 4 we restrict the cross-shelf profile to the exponential profile of Buchwald and Adams (1968) which, in addition to being a good model for the Oregon shelf, permits the cross-shelf dependence of the wave amplitudes to be solved analytically. Using this explicit cross-shelf profile we compute wave numbers which form a resonant triad, and their corresponding interaction and topography coefficients.

In Chapter 5 we investigate the purely temporal problem without alongshore topographic variation. Exact solutions for the case with no bottom friction are given in terms of Jacobian elliptic functions and their period of oscillation is computed. The problem with bottom friction is then solved numerically and compared to the frictionless solution.

In Chapter 6 we illustrate the full analytical solutions for the purely spatial problem in the presence of a piecewise constant alongshore topographic variation. In Chapter 7 the thesis is summarized and conclusions are presented.

Chapter 2

Governing Equations

2.1 Derivation of the Governing Equations

The derivation of the governing equations given in this chapter follows closely the one given in sections 3 and 4 of the paper by Hsieh and Mysak (1980). The method of multiple scales is used where here, the bottom topography is assumed not only to vary in the offshore direction, but also to vary slowly in the alongshore direction.

The dimensional nonlinear barotropic long-wave equations with bottom friction for a rotating system are

$$u^*_{t^*} + u^* u^*_{x^*} + v^* u^*_{y^*} - f v^* = -g\eta^*_{x^*} - \frac{1}{\rho} p^*_{ax^*} + \frac{1}{\rho h^*}\tau^*_1 - r^* u^* , \tag{2.1.1}$$

$$v^*_{t^*} + u^* v^*_{x^*} + v^* v^*_{y^*} + f u^* = -g\eta^*_{y^*} - \frac{1}{\rho} p^*_{ay^*} + \frac{1}{\rho h^*}\tau^*_2 - r^* v^* , \tag{2.1.2}$$

$$(h^* u^*)_{x^*} + (h^* v^*)_{y^*} = -\eta^*_{t^*} - (\eta^* u^*)_{x^*} - (\eta^* v^*)_{y^*} , \tag{2.1.3}$$

where t^* is time, x^* the offshore coordinate and y^* the alongshore coordinate; (u^*, v^*) are the horizontal velocity components in the (x^*, y^*) direction, η^* the sea surface displacement from equilibrium, p^*_a the atmospheric pressure at the sea surface, (τ^*_1, τ^*_2) the surface wind-stress components (modelled as a body force) and h^* the depth of the ocean; f, g and r^* are the Coriolis parameter, the gravitational constant and the bottom friction parameter (given by $fE_V^{1/2}$ where E_V is the "vertical" Ekman number and not included by Hsieh and Mysak 1980), respectively. We shall assume that f, r^* and the density ρ are constants.

To nondimensionalize the above equations we introduce the following dimensionless (unastericked) variables

$$\left.\begin{aligned} (x,y) &= (x^*/L, y^*/L)\,, \\ t &= ft^*\,, \\ (u,v) &= (u^*/V, v^*/V)\,, \\ \eta &= \eta^*/\eta_0\,, \\ h &= h^*/H_0\,, \\ (\tau_1,\tau_2) &= (\tau_1^*/\tau_0, \tau_2^*/\tau_0)\,, \\ p_a &= p_a^*/(\rho V L f)\,, \end{aligned}\right\} \tag{2.1.4}$$

and the following dimensionless parameters

$$\left.\begin{aligned} \varepsilon &= \frac{V}{Lf}, \text{ the Rossby number,} \\ \eta_0 &= \frac{fVL}{g}, \text{ the characteristic free surface elevation,} \\ r &= \frac{L}{V} r^*, \text{ the friction parameter,} \\ \mu^2 &= \frac{f^2 L^2}{gH}, \text{ the divergence parameter,} \end{aligned}\right\} \tag{2.1.5}$$

where the scales L, V, H, τ_0 characterize the magnitudes of horizontal length (shelf/slope region $L \approx 100$ km), horizontal velocity ($V = 10^{-1}$m s^{-1}), vertical length (shelf depth scale $H_0 = 200$ m), and surface wind-stress ($\tau_0 = 0.1$N m^{-2}). At midlatitudes f is typically 10^{-4} rad s^{-1} so that the nondimensional parameters have the following values: $\varepsilon = 0.01$, $\eta_0 = 0.1$, $r = 10^3 r^*$ and $\mu^2 = 0.05$. Physcially, the Rossby number ε is a measure of the significance of the earth's rotation for the phenomenon being described; mathematically, it is a measure of the importance of the nonlinear terms in the dynamical balance. η_0 is the geostrophic scaling for the free surface evaluation, i.e., we assume that the horizontal component of the coriolis acceleration balances the horizontal gradient of

the dynamic pressure. The divergence parameter μ^2 is a measure of the importance of vortex stretching due to free surface displacement relative to stretching due to variation in depth. Finally the friction parameter r has been nondimensionalized with respect to the advective time scale.

To focus on the effect of bottom friction and topographic variation, we shall neglect the surface wind-stress, so that (2.1.1-2.1.3) can be written in nondimensional form as follows

$$u_t + \varepsilon(uu_x + vu_y) - v = -\eta_x - p_{ax} - \varepsilon r u \,, \tag{2.1.6}$$

$$v_t + \varepsilon(uv_x + vv_y) + u = -\eta_y - p_{ay} - \varepsilon r v \,, \tag{2.1.7}$$

$$(hu)_x + (hv)_y = -\mu^2\{\eta_t + \varepsilon((\eta u)_x + (\eta v)_y)\} \,. \tag{2.1.8}$$

By forming $\partial_y(2.6)\ -\partial_x(2.7)$, the nondimensional vorticity equation can be obtained;

$$(u_y - v_x)_t + \varepsilon r(u_y - v_x) + \varepsilon[(uu_x + vu_y)_y - (uv_x + vv_y)_x] - (v_y + u_x) = 0 \,. \tag{2.1.9}$$

Making the nondivergence approximation permits us to neglect the terms on the right hand side of (2.1.8), so that we can introduce a transport streamfunction ψ such that

$$hu = \psi_y \,,\ hv = -\psi_x \,. \tag{2.1.10}$$

This approximation neglects vortex tube stretching due to surface displacements in favor of stretching due to depth variation. Its effect is to filter out Poincaré and Kelvin waves so that we can focus on continental shelf waves. Substituting (2.1.10) into (2.1.9), we obtain

$$\begin{aligned}
&\Big\{\frac{-h_y}{h^2}\psi_y + \frac{1}{h}\psi_{yy} - \frac{h_x}{h^2}\psi_x + \frac{1}{h}\psi_{xx}\Big\}_t + \varepsilon r\Big\{\frac{-h_y}{h^2}\psi_y + \frac{1}{h}\psi_{yy} - \frac{h_x}{h^2}\psi_x + \frac{1}{h}\psi_{xx}\Big\} \\
&\quad + \varepsilon\Big\{\Big(\frac{-h_x}{h^3}\psi_y^2 + \frac{1}{h^2}\psi_y\psi_{xy} + \frac{h_y}{h^3}\psi_x\psi_y - \frac{1}{h^2}\psi_x\psi_{yy}\Big)_y \\
&\quad - \Big(\frac{h_x}{h^3}\psi_x\psi_y - \frac{1}{h^2}\psi_y\psi_{xx} - \frac{h_y}{h^3}\psi_x^2 + \frac{1}{h^2}\psi_x\psi_{xy}\Big)_x\Big\} - \Big\{\frac{h_y}{h^2}\psi_x - \frac{h_x}{h^2}\psi_y\Big\} = 0 \,,
\end{aligned} \tag{2.1.11}$$

which can be rearranged into the form

$$
\begin{aligned}
h[\psi_{xxt}+\psi_{yyt}]-h_x[\psi_{xt}-\psi_y]-h_y[\psi_{yt}+\psi_x]+\varepsilon\{r(h\psi_{xx}+h\psi_{yy}-h_x\psi_x-h_y\psi_y)\\
+(\frac{h_{xy}}{h}-\frac{3h_yh_x}{h^2})\psi_x^2+(\frac{3(h_x^2-h_y^2)}{h^2}-\frac{(h_{xx}-h_{yy})}{h})\psi_x\psi_y\\
-(\frac{h_{xy}}{h}-\frac{3h_yh_x}{h^2})\psi_y^2-\frac{2h_x}{h}\psi_y\psi_{yy}+\frac{3h_y}{h}\psi_x\psi_{yy}\\
-\frac{h_y}{h}\psi_y\psi_{xy}+\frac{h_x}{h}\psi_x\psi_{xy}-\frac{3h_x}{h}\psi_y\psi_{xx}+\frac{2h_y}{h}\psi_x\psi_{xx}\\
+\psi_y\psi_{xyy}-\psi_x\psi_{yyy}-\psi_x\psi_{yxx}+\psi_y\psi_{xxx}\}=0. \qquad (2.1.12)
\end{aligned}
$$

Since we are looking for waves trapped near the coast, we impose the boundary condition $\psi \to 0$ as $x \to \infty$. To assure that there is no mass flux through the coast we impose the following boundary condition $hu = 0$ at $x = 0$. From Eq. (2.1.10) the second boundary condition becomes $\psi_y = 0$ at $x = 0$.

Next we assume that h and ψ can be expanded as

$$h(x,y,Y) = h_0(x) + \varepsilon h_1(x,Y) \qquad (2.1.13)$$

and

$$\psi(x,y,Y,t,T) = \psi^{(0)}(x,y,Y,t,T) + \varepsilon\psi^{(1)}(x,y,Y,t,T) + \ldots \qquad (2.1.14)$$

where Y and T are the "slow" variables defined by

$$\left.\begin{aligned} Y &= \varepsilon y \\ T &= \varepsilon t. \end{aligned}\right\} \qquad (2.1.15)$$

Thus in Eq. (2.1.12),

$$\left.\begin{aligned} \frac{\partial}{\partial y} &\to \frac{\partial}{\partial y}+\frac{\varepsilon\partial}{\partial Y} \\ \frac{\partial}{\partial t} &\to \frac{\partial}{\partial t}+\frac{\varepsilon\partial}{\partial T}. \end{aligned}\right\}$$

We note here, that Hsieh and Mysak assumed h to be a function of x only. By choosing h to be of the form (2.1.13), our objective is to derive wave-wave interaction equations where the effects of alongshore topographically-induced vortex tube stretching are as important as the nonlinear interactions.

Substituting (2.1.13) and (2.1.14) into (2.1.12), we obtain the $O(1)$ and the $O(\varepsilon)$ problems:

$$O(1): \begin{cases} h_0(\psi^{(0)}_{xxt} + \psi^{(0)}_{yyt}) + h_0'(\psi^{(0)}_{y} - \psi^{(0)}_{xt}) = 0 & (2.1.16) \\ \psi^{(0)} \to 0 \quad \text{as} \quad x \to \infty & (2.1.17) \\ \psi^{(0)}_{y} = 0 \quad \text{as} \quad x = 0 & (2.1.18) \end{cases}$$

and

$$O(\varepsilon): \begin{cases} h_0[\psi^{(1)}_{xxt} + \psi^{(1)}_{yyt}] + h_0'[\psi^{(1)}_{y} - \psi^{(1)}_{xt}] = r(h_0\psi^{(0)}_{xx} + h_0\psi^{(0)}_{yy} - h_0'\psi^{(0)}_{x}) \\ -h_o[\psi^{(0)}_{xxT} + \psi^{(0)}_{yyT} + 2\psi^{(0)}_{yYt}] + h_0'[\psi^{(0)}_{xT} - \psi^{(0)}_{Y}] + \left[\frac{h_0''}{h_0} - \left(\frac{h_0'}{h_0}\right)^2\right]\psi^{(0)}_{x}\psi^{(0)}_{y} \\ +\frac{3h_0'}{h_0}\psi^{(0)}_{xx}\psi^{(0)}_{y} - \frac{h_0'}{h_0}\psi^{(0)}_{x}\psi^{(0)}_{xy} - \psi^{(0)}_{xxx}\psi^{(0)}_{y} + \psi^{(0)}_{x}\psi^{(0)}_{xxy} + \frac{2h_0'}{h_0}\psi^{(0)}_{yy}\psi^{(0)}_{y} \\ -\psi^{(0)}_{xyy}\psi^{(0)}_{y} + \psi^{(0)}_{x}\psi^{(0)}_{yyy} - h_1[\psi^{(0)}_{yyt} + \psi^{(0)}_{xxt}] - h_{1x}[\psi^{(0)}_{y} - \psi^{(0)}_{xt}] & (2.1.19) \\ \\ \psi^{(1)} \to 0 \quad \text{as} \quad x \to \infty & (2.1.20) \\ \psi^{(0)}_{Y} + \psi^{(1)}_{y} = 0 \quad \text{at} \quad x = 0 & (2.1.21) \end{cases}$$

We can let $\psi^{(0)}$ have the form

$$\psi^{(0)} = A(Y,T)\phi(x)e^{i(ky-\omega t)} + c.c., \qquad (2.1.22)$$

where $c.c.$ denotes the complex conjugate of the previous term. By substituting (2.1.22) into (2.1.16), we can deduce that $\phi(x)$ must satisfy the following Sturm-Liouville problem:

$$-(\frac{-1}{h_0}\phi')' + \frac{-k^2}{h_0}\phi - \frac{1}{c}(\frac{h_0'}{h_0^2})\phi = 0\,, \qquad (2.1.23)$$

$$\phi(x) \to 0 \quad \text{as} \quad x \to \infty\,, \qquad (2.1.24)$$

$$\phi(0) = 0\,, \qquad (2.1.25)$$

where $c^{-1} = k/\omega$ is the eigenvalue (Hsieh and Mysak 1980, Boyce and DiPrima 1968, p. 493). Equation (2.1.23) with an exponential shelf profile has been solved by Buchwald and Adams (1968).

2.2 Derivation of the Amplitude Equations

In this section we derive the interaction equations for three shelf waves with frequencies ω_j and wavenumbers k_j $(j = 1, 2, 3)$. Thus $\psi^{(0)}$ has the form

$$\psi^{(0)} = \sum_{j=1}^{3} A_j(Y,T)\phi_j(x)exp[i(k_j y - \omega_j t)] + c.c. \tag{2.2.1}$$

where each of the ϕ_j's satisfies a Sturm-Liouville problem of the form (2.1.23-2.1.25):

$$-(\frac{-1}{h_0}\phi_j')' + \frac{-k_j^2}{h_0}\phi_j - \frac{1}{c_j}(\frac{h_0'}{h_0^2})\phi_j = 0\,, \tag{2.2.2}$$

$$\phi_j \to 0 \quad \text{as} \quad x \to \infty\,, \tag{2.2.3}$$

$$\phi_j = 0 \quad \text{at} \quad x = 0\,. \tag{2.2.4}$$

The phase speed of the jth wave is given by $c_j = \omega_j/k_j$. Substituting (2.2.1) into (2.1.19) and using (2.2.2) we obtain

$$\begin{aligned}
&h_0[\psi^{(1)}_{xxt} + \psi^{(1)}_{yyt}] + h_0'[\psi^{(1)}_{y} - \psi^{(1)}_{xt}] = \sum_{j=1}^{3}(\frac{-h_0'}{c_j}\phi_j A_{jT} - (h_0' + 2k_j\omega_j h_0)\phi_j A_{jY} \\
&+ (rh_0\phi_j'' - rh_0k_j^2\phi_j - rh_0'\phi_j' - ih_1\omega_j k_j^2\phi_j + ih_1\omega_j\phi_j'' - ik_j h_{1x}\phi_j \\
&+ i\omega_j h_{1x}\phi_j')A_j)exp[i(k_j y - \omega_j t)] \\
&+ \sum_{\ell=1}^{3}\sum_{m=1}^{3}(\frac{-2h_0'}{h_0}k_\ell^2\phi_\ell\phi_m + (\frac{h''}{h_0} - 3(\frac{h_0'}{h_0})^2 + k_\ell^2 - k_m^2)\phi_\ell'\phi_m + \frac{3h_0'}{h_0}\phi_\ell''\phi_m - \frac{h_0'}{h_0}\phi_\ell'\phi_m' \\
&- \phi_\ell'''\phi_m)(ik_m A_\ell A_m exp\{i[(k_\ell + \ell_m)y - (\omega_\ell + \omega_m)t]\} - ik_m A_\ell A_m^* exp\{i[(k_\ell - k_m)y \\
&- (\omega_\ell - \omega_m)t]\}) + c.c.\,.
\end{aligned} \tag{2.2.5}$$

The required wave-wave interaction equations are obtained by exploiting certain solvability conditions associated with the particular solutions to (2.2.5). We begin by observing that the frequencies and wavenumbers on the right hand side must match those on the left hand side. Thus the contribution of the particular solution to $\psi^{(1)}$ must be of the form

$$\psi^{(1)} = \sum_{\alpha} A_{\alpha}^{(1)}(Y,T)\phi_{\alpha}^{(1)}(x)exp[i(k_{\alpha}y - \omega_{\alpha}t)] + c.c. \tag{2.2.6}$$

In order to have resonant interactions the set of values for $(k_{\alpha},\omega_{\alpha})$ must be given by

$$\{(k_{\alpha},\omega_{\alpha})\} = \{(k_j,\omega_j)|\ j = 1,2,3\} \cup \{(k_{\ell},\omega_{\ell}) \pm (k_m,\omega_m)|\ \ell = 1,2,3, m = 1,2,3\}\ . \tag{2.2.7}$$

For convenience, we establish the following convention for indexing: for $\alpha = 1,2,3$, $(k_{\alpha},\omega_{\alpha}) \equiv (k_j,\omega_j)$.

We now examine the situation for $\alpha = j = 1,2,3$. Substituting (2.2.6) into the left hand side of (2.2.5), we find that for each j, $\phi_j^{(1)}$ must satisfy an inhomogeneous equation

$$-(\frac{-1}{h_0}\phi_j^{(1)\prime})' + \frac{-k_j^2}{h_0}\phi_j^{(1)} - \frac{1}{c_j}(\frac{h_0'}{h_0^2})\phi_j^{(1)} = f_j \tag{2.2.8}$$

where f_j denotes the forcing at frequency ω_j and wavenumber k_j from the right side of (2.2.5). In nonresonant cases (the conditions for resonance will be given momentarily) only the linear terms on the right side of (2.2.5) can contribute to the f_j's ($j = 1,2,3$);

$$\begin{aligned} f_j =&[\frac{h_0'}{c_j}\phi_j A_{jT}^{(0)} + (h_0' + 2k_j\omega_j h_0)\phi_j A_{jY} - (rh_0\phi_j'' - rh_0k_j^2\phi_j - rh_0'\phi_j - ih_1\omega_j k_j^2\phi_j \\ &+ ih_1\omega_j\phi_j'' - ik_j h_{1x}\phi_j + i\omega_j h_{1x}\phi_j')A_j^{(0)}]/[i\omega_j h_0^2 A_j^{(1)}]\ . \end{aligned} \tag{2.2.9}$$

However, if any of the resonance conditions are satisfied, viz.

$$\pm(\omega_1,k_1) \pm (\omega_2,k_2) \pm (\omega_3,k_3) = (0,0)\ , \tag{2.2.10}$$

some of the nonlinear terms on the right side of (2.2.5) are also of frequencies ω_j and wavenumbers k_j $(j = 1, 2, 3)$, and hence can contribute to the forcing terms f_j's. With no loss of generality, we can write the resonance conditions as

$$\omega_1 + \omega_2 + \omega_3 = 0, \quad k_1 + k_2 + k_3 = 0\,, \tag{2.2.11}$$

where $\{\omega_j\}$ and $\{k_j\}$ can have positive or negative values.

The Fredholm alternate theorem from Sturm-Liouville theory (see, e.g. Boyce and Diprima 1969, p. 506) states that the inhomogeneous problem (2.2.8) has a solution $\phi_j^{(1)}$ only if the forcing term f_j is orthogonal to the homogeneous solution ϕ_j, i.e.,

$$\int_0^\infty f_j \phi_j dx = 0\,. \tag{2.2.12}$$

When the resonance condition (2.2.11) are satisfied, the Fredholm condition (2.2.12) yields

$$\begin{aligned} &A_{jT}\int_0^\infty \frac{h_0'}{h_0^2}\phi_j^2 dx + A_{jY}[c_j\int_0^\infty \frac{h_0'}{h_0^2}\phi_j^2 + 2c_j^2\frac{1}{h_0}\phi_j^2 dx] - A_j r_j \\ &- A_j[i(\omega_j k_j^2\int_0^\infty \frac{h_1}{h_0^2}\phi_j^2 + \omega_j\frac{h_1}{h_0^2}\phi_j''\phi_j - k_j\frac{h_{1x}}{h_0^2}\phi_j^2 + \omega_j\frac{h_{1x}}{h_0^2}\phi_j'\phi_j dx)] \\ &= ic_j(K_{j\ell m} + K_{jm\ell})A_\ell^* A_m^*\,, \end{aligned} \tag{2.2.13}$$

where (j, ℓ, m) are any of the cyclic permutations of $(1, 2, 3)$ and the terms on the right-hand side of (2.2.13) arise from the nonlinear terms in (2.2.5) with

$$\begin{aligned} K_{j\ell m} = \int_0^\infty dx \frac{1}{h_0^2}k_m\phi_j\{&\frac{-2h_0'}{h_0}k_\ell^2\phi_\ell\phi_m + [\frac{h_0''}{h_0} - 3(\frac{h_0'}{h_0})^2 + k_\ell^2 - k_m^2]\phi_\ell'\phi_m \\ &+ \frac{3h_0'}{h_0}\phi_\ell''\phi_m - \frac{h_0'}{h_0}\phi_\ell'\phi_m' + \phi_\ell'\phi_m'' - \phi_\ell'''\phi_m\}, \end{aligned} \tag{2.2.14}$$

and

$$r_j = -\int_0^\infty r[\frac{-h_0'}{h_0^2}\phi_j'\phi_j + \frac{1}{h_0}\phi_j''\phi_j - \frac{k_j^2}{h_0}\phi_j^2]dx. \tag{2.2.15}$$

With repeated integration by parts and the help of the boundary conditions (2.2.3) and (2.2.4), $K_{j\ell m}$ can be re-expressed in another form

$$K_{j\ell m} = \int_0^\infty dx \frac{k_m}{h_0^2}[\frac{-3h_0'}{h_0}\phi_j'\phi_\ell'\phi_m + 2\phi_j'\phi_\ell''\phi_m + \phi_j''\phi_\ell'\phi_m - \frac{2h_0'}{h_0}k_\ell^2\phi_j\phi_\ell\phi_m + (k_\ell^2 - k_m^2)\phi_j\phi_\ell'\phi_m]. \tag{2.2.16}$$

The integrand in the integral for r_j can be simplified. By rearranging (2.2.2) and multiplying through by ϕ_j we obtain

$$\frac{-h_0'}{h_0^2}\phi_j'\phi_j + \frac{1}{h_0}\phi_j''\phi_j - \frac{k_j^2}{h_0}\phi_j^2 = \frac{1}{c_j}\frac{h_0'}{h_0^2}\phi_j^2\,, \tag{2.2.17}$$

where we recognize the left hand side to be the integrand, so that substitution into (2.2.16) yields

$$r_j = \frac{-r}{c_j}\int_0^\infty \frac{h_0'}{h_0^2}\phi_j^2 dx. \tag{2.2.18}$$

Without loss of generality, we can normalize $\{\phi_j\}$ so that

$$\int_0^\infty \frac{h_0'}{h_0^2}\phi_j^2 dx = 1. \tag{2.2.19}$$

Equation (2.2.13) then gives us the amplitude equation for the jth wave

$$(\partial_T + c_{g_j}\partial_Y)A_j = -iK_jA_\ell^*A_m^* - r_jA_j - i\mu_j(Y)A_j \tag{2.2.20}$$

where

$$K_j \equiv c_j(K_{j\ell m} + K_{jm\ell}), \quad \text{the coupling coefficient,} \tag{2.2.21}$$

$$c_{g_j} \equiv c_j(1 + \gamma_j c_j k_j^2), \quad \text{the group velocity,} \tag{2.2.22}$$

$$\gamma_j \equiv \int_0^\infty \frac{1}{h_0}\phi_j^2 dx, \tag{2.2.23}$$

$$r_j \equiv \frac{-r}{c_j}, \quad \text{the damping coefficient,} \tag{2.2.24}$$

$$\mu_j(Y) \equiv \int_0^\infty [\omega_j k_j^2 \frac{h_1}{h_0^2}\phi_j^2 - \omega_j \frac{h_1}{h_0^2}\phi_j''\phi_j + k_j\frac{h_{1x}}{h_0^2}\phi_j^2 - \omega_j\frac{h_{1x}}{h_0^2}\phi_j'\phi_j]dx, \tag{2.2.25}$$

To see that c_{gj} is indeed the group velocity, refer to Appendix 1. The last two terms on the right side of (2.2.17) are the extensions made to the theory of Hsieh and Mysak to include the effect of bottom friction and alongshore topographic variation.

It will be convenient to recast the interaction equations (2.2.17) into "standard" form. To this end we first define

$$\sigma_j(Y) = \int_0^{\epsilon y} \mu_j(\xi) d\xi \,, \tag{2.2.26}$$

and then use this definition to define the new amplitude functions $\alpha_j(Y,T)$ given by

$$\alpha_j(Y,T) = A_j(Y,T) exp[-i\sigma_j(Y)/c_{g_j}] \,, \tag{2.2.27}$$

for $j = 1,2,3$. Substitution of (2.2.27) into (2.2.20) yields the interaction equations in the form

$$\left.\begin{aligned}
(\partial_T + c_{g_1}\partial_Y)\alpha_1 &= -iK_1\alpha_2^*\alpha_3^* exp[-i\sigma_0(Y)] - r_1\alpha_1 \,, \\
(\partial_T + c_{g_2}\partial_Y)\alpha_2 &= -iK_2\alpha_3^*\alpha_1^* exp[-i\sigma_0(Y)] - r_2\alpha_2 \,, \\
(\partial_T + c_{g_3}\partial_Y)\alpha_3 &= -iK_3\alpha_1^*\alpha_2^* exp[-i\sigma_0(Y)] - r_3\alpha_3 \,,
\end{aligned}\right\} \tag{2.2.28}$$

where

$$\sigma_0(Y) = \frac{\sigma_1(Y)}{c_{g_1}} + \frac{\sigma_2(Y)}{c_{g_2}} + \frac{\sigma_3(Y)}{c_{g_3}} \,. \tag{2.2.29}$$

Chapter 3

Conservation Theorems

3.1 Conservation of Energy

In this section we will show that if we ignore friction, the total energy of the resonant triad is conserved even though energy is transferred between the waves. We start with the full vorticity equation in terms of the transport streamfunction (2.1.12) which can be written in the following form

$$[\nabla\cdot(\frac{1}{h}\nabla\psi)]_t + J[\frac{\varepsilon}{h}\nabla\cdot(\frac{1}{h}\nabla\psi) - \frac{1}{h},\psi] = 0\,, \tag{3.1.1}$$

where $J(A,B) \equiv A_xB_y - A_yB_x$. Multiplying (3.1.1) by $-\psi$ and rearranging terms gives

$$[\frac{1}{2h}\nabla\psi\cdot\nabla\psi]_t + \nabla\cdot[-\frac{\psi}{h}\nabla\psi_t + (\widehat{e}_3\times\nabla\psi)\psi(\frac{\varepsilon}{h}\nabla(\frac{1}{h}\nabla\psi) - \frac{1}{h})] = 0\,. \tag{3.1.2}$$

Equation (3.1.2) is the required energy conservation law expressed in the form (Pedlosky, 1979, Sec. 3.27)

$$E_t + \nabla\cdot\mathbf{F} = 0 \tag{3.1.3}$$

with

$$E = \frac{1}{2h}\nabla\psi\cdot\nabla\psi, \text{ the energy density,} \tag{3.1.4}$$

$$\mathbf{F} = -\frac{\psi}{h}\nabla\psi_t + \widehat{e}_3\times\nabla\psi\psi(\frac{\varepsilon}{h}\nabla(\frac{1}{h}\nabla\psi) - \frac{1}{h}), \text{ the energy flux.} \tag{3.1.5}$$

When the perturbation expansions (2.1.13) and (2.1.14) are substituted in (3.1.3), an expression of the form

$$(E_t^{(0)} + F_{1,x}^{(0)} + F_{2,y}^{(0)}) + \varepsilon(E_T^{(0)} + E_t^{(1)} + F_{1,x}^{(1)} + F_{2,Y}^{(0)} + F_{2,y}^{(1)}) + \cdots = 0 \tag{3.1.6}$$

is obtained. The leading order energy density and energy flux components are

$$E^{(0)} = \frac{1}{2h_0}\nabla\psi^{(0)}\cdot\nabla\psi^{(0)}\,, \tag{3.1.7}$$

$$F_1^{(0)} = \frac{\psi_y^{(0)}\psi^{(0)}}{h_0} - \frac{\psi^{(0)}\psi_{xt}^{(0)}}{h_0} , \tag{3.1.8}$$

$$F_2^{(0)} = -\frac{\psi_x^{(0)}\psi^{(0)}}{h_0} - \frac{\psi^{(0)}\psi_{yt}^{(0)}}{h_0} . \tag{3.1.9}$$

We now introduce the phase averaging operator (Swaters, 1986),

$$\langle(\star)\rangle = \frac{1}{(2\pi)^3}\int_0^{2\pi}\int_0^{2\pi}\int_0^{2\pi}(\star)d\theta_1 d\theta_2 d\theta_3 , \tag{3.1.10}$$

where $\theta_j = k_j y - \omega_j t$, $j = 1,2,3$ are the phase variables. Thus

$$\langle E^{(0)}\rangle = \frac{1}{(2\pi)^3}\int_0^{2\pi}\int_0^{2\pi}\int_0^{2\pi} E^{(0)}d\theta_1 d\theta_2 d\theta_3 \tag{3.1.11}$$

is interpreted as the average energy density over the plane wave phases. We note that if $\langle(\star)\rangle$ is operated on a function which is periodic with period 2π in any of the phase variables the result will vanish. If we substitute expression (2.2.1) into (3.1.7), (3.1.8), (3.1.9) we obtain

$$\begin{aligned} E^{(0)} = {} & \frac{1}{h_0}(\phi_1'^2 + k_1^2\phi_1^2)|A_1|^2 + \frac{1}{h_0}(\phi_2'^2 + k_2^2\phi_2^2)|A_2|^2 + \frac{1}{h_0}(\phi_3'^2 + k_3^2\phi_3^2)|A_3|^2 \\ & + \{\text{terms that will phase average to zero}\} , \end{aligned} \tag{3.1.12}$$

$$F_1^{(0)} = \{\text{terms that will phase average to zero}\} , \tag{3.1.13}$$

$$\begin{aligned} F_2^{(0)} = {} & \frac{1}{h_0}(2|A_1|^2\phi_1'\phi_1 + 2|A_2|^2\phi_2'\phi_2 + 2|A_3|^2\phi_3'\phi_3 - 2k_1\omega_1|A_1|^2\phi_1^2 \\ & - 2k_2\omega_2|A_2|^2\phi_2^2 - 2k_3\omega_3|A_3|^2\phi_3) \\ & + \{\text{terms that will phase average to zero}\} . \end{aligned} \tag{3.1.14}$$

Thus the kinetic energy per length is

$$\begin{aligned} KE/\text{length} = {} & \int_0^\infty \langle E^{(0)}\rangle dx \\ = {} & \int_0^\infty \frac{1}{h_0}[|A_1|^2(\phi_1'^2 + k_1^2\phi_1^2) + |A_2|^2(\phi_2'^2 + k_2^2\phi_2^2) \\ & + |A_3|^2(\phi_3'^2 + k_3^2\phi_3^2)]dx . \end{aligned} \tag{3.1.15}$$

Integrating $\frac{1}{h_0}\phi_j'^2$, $j = 1,2,3$ by parts, then invoking (2.2.2), (2.2.3), (2.2.4), and the normalization condtion (2.2.19), we obtain

$$KE/\text{length} = \frac{-1}{c_1}|A_1|^2 - \frac{1}{c_2}|A_2|^2 - \frac{1}{c_3}|A_3|^2 . \tag{3.1.16}$$

Similarly, the kinetic energy flux per length is

$$\begin{aligned}\text{Flux/length} &= \left(\int_0^\infty \langle F_1^{(0)}\rangle dx,\ \int_0^\infty \langle F_2^{(0)}\rangle dx\right) \\ &= \left(0,\ \int_0^\infty 2|A_1|^2\frac{1}{h_0}(\phi_1'\phi_1 - k_1\omega_1\phi_1^2) + 2|A_2|^2\frac{1}{h_0}(\phi_2'\phi_2 - k_2\omega_2\phi_2^2)\right. \\ &\quad \left. + 2|A_3|^2\frac{1}{h_0}(\phi_3'\phi_3 - k_3\omega_3\phi_3^2)dx\right) .\end{aligned} \tag{3.1.17}$$

Integrating $\frac{1}{h_0}\phi_j\phi_j'$ by parts, then involing (2.2.2), (2.2.3) (2.2.4), and the normalization condition (2.2.19), we obtain

$$\begin{aligned}&\text{Flux/length} \\ &= (0,\ |A_1|^2(1 - 2k_1^2c_1\gamma_1) + |A_2|^2(1 - 2k_2^2c_2\gamma_2) + |A_3|^2(1 - 2k_3^2c_3\gamma_3))\end{aligned} \tag{3.1.18}$$

where γ_j is defined in (2.2.23). Using the expression for the group velocity (2.2.22) the energy flux per length is given by

$$\text{Flux/length} = \left(0,\ \frac{c_{g1}}{c_1}|A_1|^2 + \frac{c_{g2}}{c_2}|A_2|^2 + \frac{c_{g3}}{c_3}|A_3|^2\right) . \tag{3.1.19}$$

To show that energy is conserved we go back to the three wave equations (2.2.20) written out explicitely without bottom friction

$$\left.\begin{aligned}(\partial_T + c_{g1}\partial_Y)A_1 &= -iK_1A_2^*A_3^* - i\mu_1(Y)A_1 ,\\ (\partial_T + c_{g2}\partial_Y)A_2 &= -iK_2A_3^*A_1^* - i\mu_2(Y)A_2 ,\\ (\partial_T + c_{g3}\partial_Y)A_3 &= -iK_3A_1^*A_2^* - i\mu_3(Y)A_3 .\end{aligned}\right\} \tag{3.1.20}$$

If we multiply the jth equation of (3.1.20) by $\frac{-A_j^*}{c_j}$ we obtain the following system

$$\left.\begin{aligned}
\frac{-A_1^*}{c_1}(\partial_T + c_{g1}\partial_Y)A_1 &= \frac{iK_1}{c_1}A_1^*A_2^*A_3^* + \frac{i\mu_1(Y)}{c_1}|A_1|^2\,, \\
\frac{-A_2^*}{c_2}(\partial_T + c_{g2}\partial_Y)A_2 &= \frac{iK_2}{c_2}A_1^*A_2^*A_3^* + \frac{i\mu_2(Y)}{c_2}|A_2|^2\,, \\
\frac{-A_3^*}{c_3}(\partial_T + c_{g3}\partial_Y)A_3 &= \frac{iK_3}{c_3}A_1^*A_2^*A_3^* + \frac{i\mu_3(Y)}{c_3}|A_3|^2\,.
\end{aligned}\right\} \tag{3.1.21}$$

Adding the equations of (3.1.21) along with their complex conjugates we arrive at the energy equation

$$\begin{aligned}
\partial_T(\frac{-1}{c_1}|A_1|^2 - \frac{1}{c_2}|A_2|^2 - \frac{1}{c_3}|A_3|^2) + \nabla\cdot(0, \frac{-c_{g1}}{c_1}|A_1|^2 - \frac{c_{g2}}{c_2}|A_2|^2 - \frac{c_{g3}}{c_3}|A_3|^2) \\
= 2Re\{iA_1A_2A_3\}(\frac{-K_1}{c_1} - \frac{K_2}{c_2} - \frac{K_3}{c_3})\,.
\end{aligned} \tag{3.1.22}$$

The conservation of the total energy is then expressed by the following theorem.

THEOREM:

$$\partial_T(\frac{-1}{c_1}|A_1|^2 - \frac{1}{c_2}|A_2|^2 - \frac{1}{c_3}|A_3|^2) + \nabla\cdot(0, \frac{-c_{g1}}{c_1}|A_1|^2 - \frac{c_{g2}}{c_2}|A_2|^2 - \frac{c_{g3}}{c_3}|A_3|^2) = 0\,. \tag{3.1.23}$$

PROOF: We will reproduce the proof given by Hsieh and Mysak (1980, Section 4) modified ot take into account variable topography. Equation (3.1.23) follows from (3.1.22) if and only if

$$-\frac{K_1}{c_1} - \frac{K_2}{c_2} - \frac{K_3}{c_3} = 0. \tag{3.1.24}$$

From (2.2.21) this amounts to proving

$$K_{123} + K_{132} + K_{231} + K_{213} + K_{312} + K_{321} = 0. \tag{3.1.25}$$

Remembering (from (2.2.16)) that

$$K_{j\ell m} = \int_0^\infty [\{\frac{3k_m h_0'}{h_0^3}\phi_j'\phi_\ell'\phi_m'\} + \frac{2k_m}{h_0^2}\phi_j'\phi_\ell''\phi_m + \frac{k_m}{h_0^2}\phi_j''\phi_\ell'\phi_m - \{\frac{2k_m k_\ell^2 h_0'}{h_0^3}\phi_j\phi_\ell\phi_m\} + \frac{(k_\ell^2 - k_m^2)k_m}{h_0^2}\phi_j\phi_\ell'\phi_m]dx,$$

we integrate the terms in curly brackets by parts to elliminate derivatives of h_0. The first term in curly brackets yields,

$$\int_0^\infty -\frac{3h_0'}{h_0^3}\phi_j'\phi_\ell'\phi_m dx = \frac{3}{2h_0^2}\phi_j'\phi_\ell'\phi_m|_0^\infty - \int_0^\infty \frac{3}{2h_0^2}[\phi_j''\phi_\ell'\phi_m' + \phi_j'\phi_\ell''\phi_m + \phi_j'\phi_\ell'\phi_m']dx\ , \tag{3.1.26}$$

While the second term in curly brackets yields,

$$\int_0^\infty -\frac{2h_0'}{h_0^3}\phi_j\phi_\ell\phi_m dx = \frac{1}{h_0^2}\phi_j\phi_\ell\phi_m|_0^\infty - \int_0^\infty [\frac{1}{h_0^2}\phi_j'\phi_\ell\phi_m + \phi_j\phi_\ell'\phi_m + \phi_j\phi_\ell\phi_m']dx, \tag{3.1.27}$$

On account of the boundary conditions, the first term on the right hand side of (3.1.26) and (3.1.27) vanishes. Thus (2.1.16) can be expressed, after grouping like terms, as follows

$$K_{j\ell m} = \int_0^\infty dx\frac{1}{h_0^2}[-\frac{1}{2}k_m\phi_j''\phi_\ell'\phi_m + \frac{1}{2}k_m\phi_j'\phi_\ell''\phi_m - \frac{3}{2}k_m\phi_j'\phi_\ell'\phi_m' - k_m k_\ell^2\phi_j'\phi_\ell\phi_m - k_m k_\ell^2\phi_j\phi_\ell\phi_m' - k_m^3\phi_j\phi_\ell'\phi_m]. \tag{3.1.28}$$

Using the resonance condition $k_m = -(k_j + k_\ell)$, (3.1.28) can be expressed as

$$K_{j\ell m} = \int_0^\infty dx\frac{1}{h_0^2}[-\frac{1}{2}k_m\phi_j''\phi_\ell''\phi_m + \frac{1}{2}k_m\phi_j'\phi_\ell''\phi_m - \frac{3}{2}k_m\phi_j'\phi_\ell'\phi_m' - k_m k_\ell^2\phi_j'\phi_\ell\phi_m + (k_j + k_\ell)k_\ell^2\phi_j\phi_\ell\phi_m' - k_m^3\phi_j\phi_\ell'\phi_m]. \tag{3.1.29}$$

So that we can re-express $K_{j\ell m}$ as

$$K_{j\ell m} = K_{j\ell m}^{(1)} + K_{j\ell m}^{(2)} + K_{j\ell m}^{(3)} + K_{j\ell m}^{(4)}, \tag{3.1.30}$$

where

$$K^{(1)}_{j\ell m} = \int_0^\infty dx \frac{1}{h_0^2}[\frac{1}{2}k_m\phi'_j\phi''_\ell\phi_m - \frac{1}{2}k_m\phi''_j\phi'_\ell\phi_m], \tag{3.1.31}$$

$$K^{(2)}_{j\ell m} = \int_0^\infty dx \frac{1}{h_0^2}[-\frac{3}{2}k_m\phi'_j\phi'_\ell\phi'_m] = K^{(2)}_{j\ell m}, \tag{3.1.32}$$

$$K^{(3)}_{j\ell m} = \int_0^\infty dx \frac{1}{h_0^2}[-k_m k_\ell^2\phi'_j\phi_\ell\phi_m + k_j k_\ell^2\phi_j\phi_\ell\phi'_m], \tag{3.1.33}$$

$$K^{(4)}_{j\ell m} = \int_0^\infty dx \frac{1}{h_0^2}[k_\ell^3\phi_j\phi_\ell\phi'_m - k_m^3\phi_j\phi'_\ell\phi_m]. \tag{3.1.34}$$

We note that

$$K^{(1)}_{j\ell m} + K^{(1)}_{\ell jm} = 0, \tag{3.1.35}$$

$$K^{(3)}_{j\ell m} + K^{(3)}_{m\ell j} = 0, \tag{3.1.36}$$

$$K^{(4)}_{j\ell m} + K^{(4)}_{jm\ell} = 0, \tag{3.1.37}$$

and that

$$\sum_{m=1}^{3} K^{(2)}_{j\ell m} = \sum_{m=1}^{3} K^{(2)}_{\ell jm} = (k_1 + k_2 + k_3)\int_0^\infty dx \frac{1}{h_0^2}[-\frac{3}{2}\phi'_1\phi'_2\phi'_3] = 0 \tag{3.1.38}$$

by the resonance conditions.

Therefore (3.1.25) can be written as

$$\begin{aligned} &K_{123} + K_{132} + K_{231} + K_{213} + K_{312} + K_{321} \\ &\quad = \sum_{m=1}^{3}(K^{(1)}_{j\ell m} + K^{(1)}_{\ell jm}) + \sum_{m=1}^{3}(K^{(2)}_{j\ell m} + K^{(2)}_{\ell jm}) \\ &\quad + \sum_{m=1}^{3}(K^{(3)}_{j\ell m} + K^{(3)}_{m\ell j}) + \sum_{m=1}^{3}(K^{(4)}_{j\ell m} + K^{(4)}_{jm\ell}) = 0 \end{aligned} \tag{3.1.39}$$

and the theorem for the conservation of total energy is proved.

□

3.2. The Manley-Rowe Relations

Three other quantities expressed as the Manley-Rowe relations (Craik 1985; Cree 1990) are also preserved. We begin with the amplitude equations (2.2.28) in standard form and set the friction terms to zero. We multiply each equation by $\alpha_1^*, \alpha_2^*, \alpha_3^*$ respectively and add each equation to its complex conjugate to obtain

$$(\partial_T + c_{g1}\partial_Y)|\alpha_1|^2 = -iK_1(\alpha_1^*\alpha_2^*\alpha_3^* \exp[i\sigma_0(Y)] - \alpha_1\alpha_2\alpha_3 \exp[-i\sigma_0(Y)]) , \tag{3.2.1a}$$

$$(\partial_T + c_{g2}\partial_Y)|\alpha_2|^2 = -iK_2(\alpha_1^*\alpha_2^*\alpha_3^* \exp[i\sigma_0(Y)] - \alpha_1\alpha_2\alpha_3 \exp[-i\sigma_0(Y)]) , \tag{3.2.1b}$$

$$(\partial_T + c_{g3}\partial_Y)|\alpha_3|^2 = -iK_3(\alpha_1^*\alpha_2^*\alpha_3^* \exp[i\sigma_0(Y)] - \alpha_1\alpha_2\alpha_3 \exp[-i\sigma_0(Y)]) . \tag{3.2.1c}$$

Forming the following equations

$$K_2 \times (3.2.1a) - K_1 \times (3.2.1b) ,$$

$$K_3 \times (3.2.1b) - K_2 \times (3.2.1c) ,$$

$$K_1 \times (3.2.1c) - K_3 \times (3.2.1a) ,$$

yields

$$(\partial_T + c_{g1}\partial_Y)K_2|\alpha_1|^2 - (\partial_T + c_{g2}\partial_Y)K_1|\alpha_2|^2 = 0 , \tag{3.2.2a}$$

$$(\partial_T + c_{g2}\partial_Y)K_3|\alpha_2|^2 - (\partial_T + c_{g3}\partial_Y)K_2|\alpha_3|^2 = 0 , \tag{3.2.2b}$$

$$(\partial_T + c_{g3}\partial_Y)K_1|\alpha_3|^2 - (\partial_T + c_{g1}\partial_Y)K_3|\alpha_1|^2 = 0 . \tag{3.2.2c}$$

We now integrate the above three equations and assume that each $\alpha_j \to 0$ as $Y \to \pm\infty$ which gives us

$$\partial_T \int_{-\infty}^{\infty} (K_2|\alpha_1|^2 + K_1|\alpha_2|^2) dY = 0 , \tag{3.2.3a}$$

$$\partial_T \int_{-\infty}^{\infty} (K_3|\alpha_2|^2 + K_2|\alpha_3|^2) dY = 0 , \tag{3.2.3b}$$

$$\partial_T \int_{-\infty}^{\infty} (K_1|\alpha_3|^2 + K_3|\alpha_1|^2) dY = 0 . \tag{3.2.3c}$$

Since the α_j's depend only on the slow variables Y, T, (3.2.3) reveals that the Manley-Rowe relations may be written in the following form

$$\int_{-\infty}^{\infty} (K_2|\alpha_1|^2 + K_1|\alpha_2|^2)dY = \text{constant} , \tag{3.2.4a}$$

$$\int_{-\infty}^{\infty} (K_3|\alpha_2|^2 + K_2|\alpha_3|^2)dY = \text{constant} , \tag{3.2.4b}$$

$$\int_{-\infty}^{\infty} (K_1|\alpha_3|^2 + K_3|\alpha_1|^2)dY = \text{constant} . \tag{3.2.4c}$$

Chapter 4

The Theory Applied to an Exponential Shelf Profile.

4.1 Resonant Interactions on a Shelf with an Exponential Cross Shelf Profile.

In order to proceed further in our study of the resonant triad, it is necessary to specify explicitly the cross-shelf profile $h_0(x)$. In this chapter we will use the Buchwald and Adams (1968) exponential shelf profile for h_0, i.e.,

$$h_0(x) = \begin{cases} H_1 e^{2bx}\,, & 0 \le x \le 1 \\ H_2\,, & 1 < x < \infty \end{cases} \tag{4.1.1}$$

where b, H_1 and H_2 are constant parameters with $H_2 = H_1 e^{2b}$. This is a good model for the Oregon shelf which is the shelf for which Hsieh and Mysak (1980) originally developed their wave-wave interaction theory. With this choice of h_0, Eq (2.2.2) becomes

$$\begin{cases} \phi_j'' - 2b\phi_j' - k_j^2\phi_j - \frac{2bk_j}{\omega_j}\phi_j = 0\,, & 0 \le x \le 1 \\ \phi_j'' - k^2\phi_j - j\phi_j = 0\,, & 1 < x < \infty \end{cases} \tag{4.1.2}$$

with boundary conditions,

$$\phi_j(0) = 0\,, \tag{4.1.3}$$

and

$$\phi(x) \to 0 \text{ as } x \to \infty. \tag{4.1.4}$$

Equation (4.1.2) is a second order, homogeneous, constant coefficients equation. The solution to (4.1.2) appropriate to the shelf region ($0 \le x \le 1$) and satisfying condition (4.1.3) is given by Buchwald and Adams (1968), and is

$$\phi_{sj}(x) = B_j e^{b(x-1)} \sin(\xi_j x)\,, \tag{4.1.5}$$

where B_j is an arbitrary constant and ξ_j is given by the dispersion relation

$$\omega_j = 2bk_j[\xi_j^2 + k_j^2 + b^2]. \tag{4.1.6}$$

In the deep ocean ($1 < x < \infty$), the appropriate solution that satisfies condition (4.1.4) is

$$\phi_{0j} = C_j e^{-|k_j|(x-1)}. \tag{4.1.7}$$

At the boundary between the continential shelf and the deep ocean ($x = 1$), there are two matching conditions. The first is that the normal flux $h_0 u$, must be continuous across the boundary, and thus ϕ_j must be continuous as well. Hence

$$\phi_{sj}(1) = \phi_{0j}(1)\,; \tag{4.1.8}$$

whence we obtain

$$C_j = B_j \sin(\xi_j). \tag{4.1.9}$$

The second boundary condition ensures pressure continuity across $x = 1$ and is obtained by integrating (2.1.23) from $x = 1 - \epsilon$ to $x = 1 + \epsilon$, and then taking the limit as $\epsilon \to 0$, and finally invoking condition (4.1.8) to find that

$$\phi'_{sj}(1) - \phi'_{0j}(1) = 0\,; \tag{4.1.10}$$

whence we obtain

$$B_j(b\sin(\xi_j) + \xi_j \cos(\xi_j)) = -C_j|k_j|. \tag{4.1.11}$$

For (4.1.9) and (4.1.11) to be compatible the following equation must be satisfied

$$\tan(\xi_j) = -\xi_j/(b + |k_j|). \tag{4.1.12}$$

The dispersion relation (4.1.6) and (4.1.12) can be solved numerically to obtain the different mode dispersion curves. Figure 1 shows the dispersion curves for the first four modes. Appendix 2 provides a C-language computer program to solve (4.1.6) and (4.1.12), and plot the dispersion curves.

The resonance conditions (2.2.11) can be rewritten as

$$\omega_1 + \omega_2 = \omega_3, \text{ and } k_1 + k_2 = k_3, \tag{4.1.13}$$

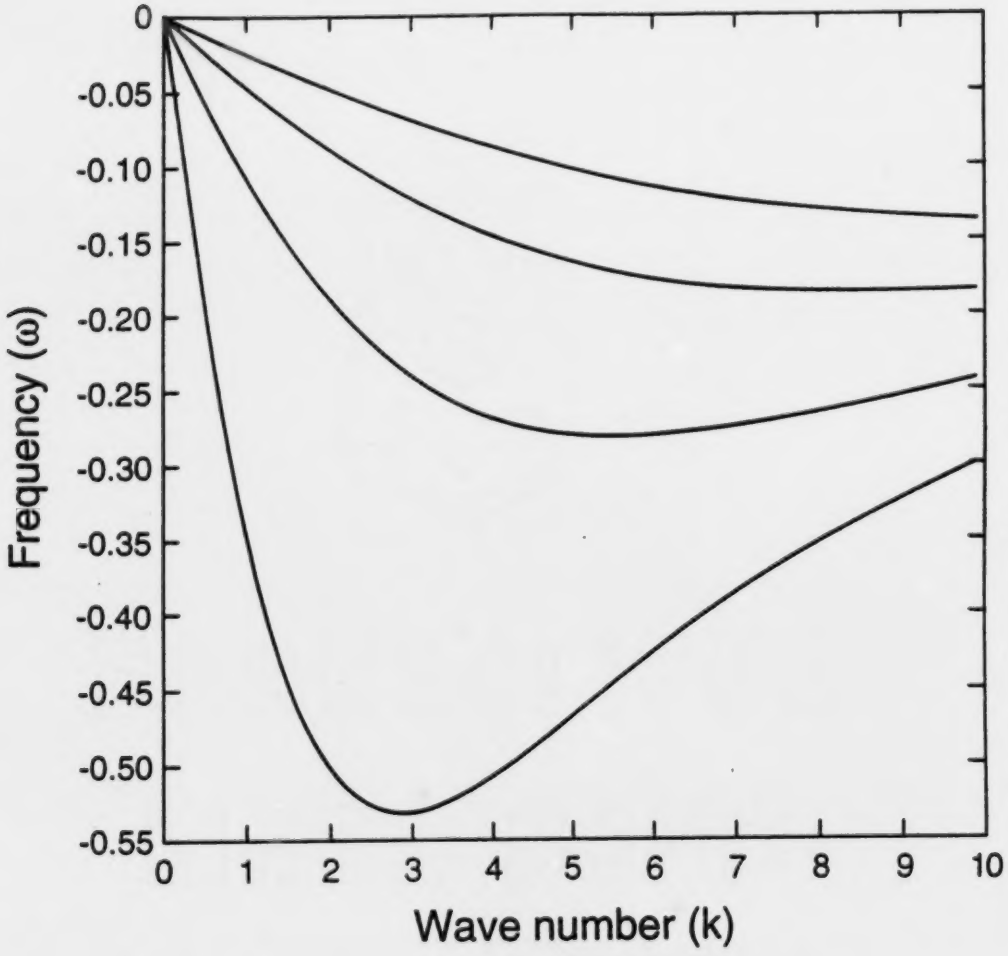


Fig. 1. Dispersion relation of the first four modes for the exponential cross shelf profile given in (4.1.1) with $H_2 = 14.2$ and $b = 1.65$.

and this last condition can also be expressed as a vector relation

$$(k_1,\omega_1)+(k_2,\omega_2)=(k_3,\omega_3). \tag{4.1.14}$$

Thus, given the frequency ω_1 and the mode of the first wave we can obtain the wave number k_1. Having fixed the first vector (k_1,ω_1), the vector equation (4.1.14) along with the dispersion curves can be used to deduce the other members of the triad, namely (k_2,ω_2) and (k_3,ω_3). Geometrically, the vector addition in (4.1.14) implies that $(0,0)$, (k_1,ω_1), (k_2,ω_2) and (k_3,ω_3) are the four vertices of a parallelogram lying in the (k,ω) plane, (Fig. 2). Appendix 3 provides a C computer program to compute a resonant triad when one is given ω_1.

Thus the solution to (4.1.2) with boundary conditions (4.1.3) and (4.1.4), and matching condition (4.1.8) for a resonant triad is

$$\phi_j^{(n)}=\begin{cases} N_j^{(n)}\sin(\xi_j^{(n)}x)e^{b(x-1)}\,, & 0\le x<1 \\ N_j^{(n)}\sin(\xi_j^{(n)})e^{-|k_j|(x-1)}\,, & 1<x<\infty \end{cases} \tag{4.1.15}$$

where the constant B_j has been replaced by $N_j^{(n)}$. The normalization constant $N_j^{(n)}$, is obtained from the normalization condition (2.2.19), and is found to be

$$N_j^{(n)}=\left[\frac{2\xi_j^{(n)}H_2}{b(2\xi_j^{(n)}-sin(2\xi_j^{(n)}))}\right]^{1/2}. \tag{4.1.16}$$

The superscript (n) denotes the nth offshore mode, and will be dropped for convenience in the next section.

4.2. Phase Velocities, Group Velocities, Interaction Coefficients, Friction Coefficients and Topography Coefficients for an Exponential Shelf.

Having obtained explicit expression for the ϕ_j's we can obtain analytical expressions for the coefficients appearing in the interaction equations (2.2.28) derived in section 2.2.

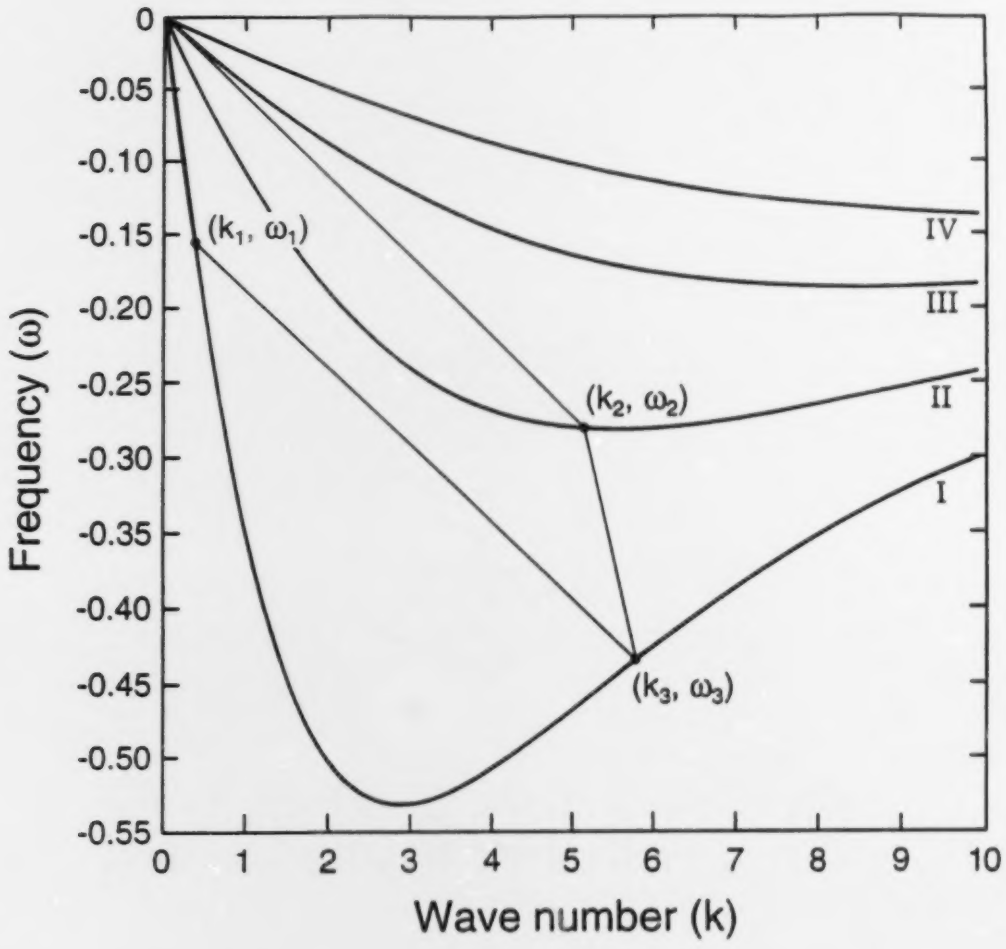


Fig. 2. The resonant triad involving waves of the lowest possible modes. With (k_1,ω_1) fixed from observation, (k_2,ω_2) and (k_3,ω_3) are then uniquely determined by constructing the parallelogram such that, with two vertices fixed at (k_1,ω_1), and $(0,0)$, the remaining two vertices touch the dispersion curves for modes I and II.

Substituting (4.1.15) into (2.2.23) and performing the integration we obtain the following expression for γ_j:

$$\gamma_j = \frac{N_j^2}{H_2}[(\frac{1}{2} - \frac{\sin(2\xi_j)}{4\xi_j}) + \frac{\sin^2(\xi_j)}{2|k_j|}] \,. \tag{4.2.1}$$

The γ_j's can then be substituted into (2.2.22) to obtain the group velocities for each resonant wave.

As shown by Hsieh and Mysak (1980), substituting (4.1.15) into (2.2.16) and performing the integration permits us to express the interaction coefficients in the following form

$$\begin{aligned} K_{j\ell m} = N_j N_\ell N_m \frac{1}{H_2^2} k_m \{ & b(-3b^2 - \xi_j^2 - 2\xi_\ell^2 - 3k_\ell^2 - k_m^2) I_1 \\ & + \xi_\ell(-b^2 - \xi_j^2 + k_\ell^2 - k_m^2) I_2 + 2\xi_j(-b^2 - \xi_\ell^2) I_3 \\ & + \frac{\sin(\xi_j)\sin(\xi_\ell)\sin(\xi_m)}{|k_j| + |k_\ell| + |k_m|}[k_\ell^2(-2|k_j| - |k_\ell|) + |k_\ell|(k_m^2 - k_j^2)]\} \end{aligned} \tag{4.2.2}$$

where

$$\begin{aligned} I_1 = 1/4[& (bA_1 + b_1B_1 - b_1C_1) - (bA_2 + b_2B_2 - b_2C_2) \\ & - (bA_3 + b_3B_3 - b_3C_3) + (bA_4 + b_4B_4 - b_4C_4)] \,, \end{aligned} \tag{4.2.3}$$

$$\begin{aligned} I_2 = 1/4[& (b_1A_1 - bB_1 + b_1C_1) - (b_2A_2 - bB_2 + b_2C_2) \\ & + (b_3A_3 - bB_3 + b_3C_3) - (b_4A_4 - bB_4 + b_4C_4)] \,, \end{aligned} \tag{4.2.4}$$

$$\begin{aligned} I_3 = 1/4[& -(b_1A_1 - bB_1 + bC_1) + (b_2A_2 - bB_2 + bC_2) \\ & + (b_3A_3 - bB_3 + bC_3) - (b_4A_4 - bB_4 + bC_4)] \,, \end{aligned} \tag{4.2.5}$$

$$\left.\begin{aligned} b_1 \equiv \xi_j - \xi_\ell - \xi_m, \; b_2 \equiv \xi_j - \xi_\ell + \xi_m \,, \\ b_3 \equiv \xi_j + \xi_\ell - \xi_m, \; b_4 \equiv \xi_j + \xi_\ell + \xi_m \,, \end{aligned}\right\} \tag{4.2.6}$$

and

$$[A_i, B_i, C_i] \equiv \frac{1}{b^2 + b_i^2}[\sin b_i, \cos b_i, e^b], \quad i = 1, 2, 3, 4. \tag{4.2.7}$$

The $K_{j\ell m}$'s can then be combined via (2.2.21) to form the K_j's appearing in (2.2.28).

If h_1 is assumed to be a function of Y only, then we can substitute equation (4.1.15) into (2.2.25) and perform the integration to obtain the following expression for the topography coefficients

$$\mu_j(Y) = h_1(Y)\mu_{j0}, \tag{4.2.8}$$

where

$$\mu_{j0} = \frac{N_j^2\omega_j}{4H_2^2}[(b^2 - k_j^2 - \xi_j^2)/b + e^{2b}(-b^2 + k^2 - z^2)/b + \{be^{2b}(b^2 - k_j^2 - 3\xi_j^2) + b(-b^2 + k_j^2 + 3\xi_j^2)\cos(2\xi_j) + \xi_j(3b^2 - k_j^2 - \xi_j^2)\sin(2\xi_j)\}/(b^2 + \xi_j^2)]. \tag{4.2.9}$$

Using the parameters for the Oregon shelf given in Hsieh and Mysak (1980), ($H_1 = 0.524$, $H_2 = 14.2$, $b = 1.65$), Tables 1, 2 and 3 gives combinations of wave numbers and frequencies which form resonant triads, along with the frequencies (ω_j), the phase velocities (c_j), the group velocities (c_{gj}), the interaction coefificents K_j, the friction coefficients (r_j) and the topography coefficients (μ_{j0}). Appendix 4 provides a C computer program to produce Tables 1, 2, and 3.

Table 1. Table of wavenumbers, frequencies and coefficients for a resonant triad for the exponential cross shelf profile given by (4.1.1) with $H_2 = 14.2$ and $b = 1.65$, and with damping parameter $r = 2.8$.

j	ξ_j	k_j	ω_j	c_j	c_{g_j}	r_j	K_j	μ_{j0}
1	2.295	0.382	-0.1549	-0.4055	-0.3734	6.904	-8.757	0.2796
2	5.609	5.365	-0.2812	-0.05241	-0.001292	5.342	-2.054	0.7497
3	2.782	-5.747	0.4361	-0.07589	0.0416	36.9	4.613	0.5446

Table 2. Table of wavenumbers, frequencies and coefficients for a resonant triad for the exponential cross shelf profile given by (4.1.1) with $H_2 = 14.2$ and $b = 16.5$, and with damping parameter $r = 0.0028$.

j	ξ_j	k_j	ω_j	c_j	c_{g_j}	r_j	K_j	μ_{j0}
1	2.295	0.382	-0.1549	-0.4055	-0.3734	0.006904	-8.757	0.2796
2	5.609	5.365	-0.2812	-0.05241	-0.001292	0.05342	-2.054	0.7497
3	2.782	-5.747	0.4361	-0.07589	0.0416	0.0369	4.613	0.5446

Table 3. Table of wavenumbers, frequencies and coefficients for a resonant triad for the exponential cross shelf profile given by (4.1.1) with $H_2 = 14.5$ and $b = 1.65$, and with damping parameter $r = 2.8$.

j	ξ_j	k_j	ω_j	c_j	c_{g_j}	r_j	K_j	μ_{j0}
1	2.405	1	-0.3472	-0.3472	-0.2468	8.065	-5.078	0.6386
2	5.307	1.938	-0.1846	-0.09526	-0.06788	29.39	-0.741	1.06
3	2.622	-2.938	0.5318	-0.181	0.0027839	15.47	4.056	-0.6983
1	2.406	1.01	-0.3496	-0.3462	-0.2447	8.088	-5.054	0.6431
2	5.303	1.903	-0.1822	-0.09576	-0.06892	29.24	-0.7286	1.049
3	2.62	-2.913	0.5319	-0.1826	0.001383	15.34	4.055	-0.7054
1	2.408	1.02	-0.3521	-0.3452	-0.2427	8.112	-5.03	0.6475
2	5.299	1.868	-0.1798	-0.09625	-0.06996	29.09	-0.7163	1.037
3	2.618	-2.888	0.5319	-0.1842	9.533e-06	15.2	4.054	-0.7123
1	2.409	1.03	-0.3545	-0.3442	-0.2406	8.135	-5.007	0.6518
2	5.294	1.834	-0.1774	-0.09673	-0.07099	28.95	-0.7041	1.025
3	2.616	-2.864	0.5319	-0.1857	-0.001381	15.08	4.054	-0.719
1	2.411	1.04	-0.3569	-0.3432	-0.2386	8.159	-4.984	0.6561
2	5.29	1.799	-0.1749	-0.09721	-0.07201	28.8	-0.692	1.013
3	2.614	-2.839	0.5318	-0.1873	-0.002787	14.95	4.054	-0.7255
1	2.412	1.05	-0.3593	-0.3422	-0.2365	8.183	-4.962	0.6604
2	5.286	1.765	-0.1725	-0.09769	-0.07303	28.66	-0.68	1.001
3	2.612	-2.815	0.5317	-0.1889	-0.00421	14.83	4.053	-0.7317
1	2.414	1.06	-0.3616	-0.3412	-0.2345	8.207	-4.94	0.6646
2	5.282	1.732	-0.17	-0.09816	-0.07404	28.53	-0.6681	0.9887
3	2.61	-2.792	0.5316	-0.1904	-0.005648	14.7	4.053	-0.7377
1	2.415	1.07	-0.364	-0.3401	-0.2324	8.232	-4.919	0.6687
2	5.278	1.699	-0.1675	-0.09862	-0.07504	28.39	-0.6563	0.976
3	2.608	-2.769	0.5315	-0.192	-0.007102	14.59	4.054	-0.7436
1	2.417	1.08	-0.3663	-0.3391	-0.2304	8.256	-4.898	0.6729
2	5.274	1.665	-0.165	-0.09908	-0.07603	28.26	-0.6446	0.9632
3	2.606	-2.745	0.5313	-0.1935	-0.008571	14.47	4.054	-0.7492

Table 3 (continued)

j	ξ_j	k_j	ω_j	c_j	c_{g_j}	r_j	K_j	μ_{j0}
1	2.418	1.09	-0.3686	-0.3381	-0.2284	8.281	-4.877	0.6769
2	5.27	1.633	-0.1625	-0.09953	-0.07702	28.13	-0.633	0.9502
3	2.604	-2.723	0.5311	-0.195	-0.01005	14.36	4.054	-0.7546
1	2.42	1.1	-0.3708	-0.3371	-0.2263	8.306	-4.858	0.6809
2	5.265	1.6	-0.16	-0.09998	-0.07799	28.01	-0.6215	0.937
3	2.602	-2.7	0.5308	-0.1966	-0.01155	14.24	4.055	-0.7599
1	2.421	1.11	-0.3731	-0.3361	-0.2243	8.33	-4.838	0.6849
2	5.561	1.568	-0.1575	-0.1004	-0.07896	27.88	-0.61	0.9237
3	2.601	-2.678	0.5306	-0.1981	-0.01307	14.13	4.055	-0.7649
1	2.423	1.12	-0.3753	-0.3351	-0.2223	8.355	-4.819	0.6888
2	5.257	1.536	-0.1549	-0.1009	-0.07992	27.76	-0.5987	0.9102
3	2.599	-2.656	0.5303	-0.1996	-0.01459	14.03	4.056	-0.7698
1	2.424	1.13	-0.3775	-0.3341	-0.2203	8.381	-4.8	0.6927
2	5.253	1.504	-0.1524	-0.1013	-0.08087	27.65	-0.5378	0.8965
3	2.597	-2.634	0.5299	-0.2011	-0.01613	13.92	4.057	-0.7745
1	2.426	1.14	-0.3797	-0.3331	-0.2183	8.406	-4.782	0.6965
2	5.249	1.473	-0.1498	-0.1017	-0.08181	27.53	-0.5761	0.8827
3	2.595	-2.613	0.5296	-0.2027	-0.01769	13.82	4.056	-0.779
1	2.427	1.15	-0.3819	-0.3321	-0.2163	8.431	-4.746	0.7003
2	5.245	1.442	-0.1473	-0.1021	-0.08274	27.42	-0.565	0.8687
3	2.593	-2.592	0.5292	-0.2042	-0.01925	13.71	4.058	-0.7834

Chapter 5

5.1 The Purely Temporal Problem

Although there are no known solutions to the general initial value problem (2.2.25), various methods with simplifying assumptions can be used to study the behavior of the resonant triad. Cree (1990) gives an overview of the "pump-wave" approximation, the inverse scattering method and solutions in terms of a nonlinear potential. The inverse scattering method can be used to obtain solutions that depend on both time and space, but such solutions are very complex to obtain and difficult to interpret. To focus in turn on the effects of bottom friction and alongshore topographic variation, we will look, in this chapter, at the purely temporal problem with no alongshore topography variation, and in the next chapter, we will study the steady state problem in which we retain the alongshore topographic variation.

To study the purely temporal problem we start with (2.2.20) and set $\partial_Y \equiv 0$. We also neglect alongshore topographic variation by setting $\sigma_0(Y) \equiv 0$. With the above simplifications (2.2.20) reduces to

$$\left.\begin{aligned} \frac{dA_1}{dT} &= -iK_1 A_2^* A_3^* - r_1 A_1 \ , \\ \frac{dA_2}{dT} &= -iK_2 A_3^* A_1^* - r_2 A_2 \ , \\ \frac{dA_3}{dT} &= -iK_3 A_1^* A_2^* - r_3 A_3 \ . \end{aligned}\right\} \tag{5.1.1}$$

If we write

$$A_j(Y,T) \equiv a_j(Y,T)\exp[i\theta(Y,T)] \ , \tag{5.1.2}$$

where a_j and θ_j are real functions, and substitute this form into our amplitude equations (5.1.1), we obtain

$$\left.\begin{aligned} \frac{da_1}{dT} &= -K_1 a_2 a_3 \sin(\theta) - r_1 a_1 \ , \\ \frac{da_2}{dT} &= -K_2 a_3 a_1 \sin(\theta) - r_2 a_2 \ , \\ \frac{da_3}{dT} &= -K_3 a_1 a_2 \sin(\theta) - r_3 a_3 \ , \end{aligned}\right\} \tag{5.1.3}$$

and

$$\left.\begin{aligned} a_1\frac{d\theta_1}{dT} &= -iK_1a_2a_3\cos(\theta)\,, \\ a_2\frac{d\theta_2}{dT} &= -iK_2a_3a_1\cos(\theta)\,, \\ a_2\frac{d\theta_3}{dT} &= -iK_3a_1a_2\cos(\theta)\,, \end{aligned}\right\} \tag{5.1.4}$$

where $\theta = \theta_1 + \theta_2 + \theta_3$.

To obtain numerical values for the coefficients in the above equations, we use the exponential shelf profile for the Oregon shelf, and the resonant triad used by Hsieh and Mysak (1980) and Cutchin and Smith (1973), i.e.

$$\left.\begin{aligned} b &= 1.65\,, \\ H_1 &= 0.524\,, \\ H_2 &= 14.2\,, \\ (k_1,\omega_1) &= (0.382,-0.155)\,, \\ (k_2,\omega_2) &= (5.362,-0.281)\,, \\ (k_2,\omega_3) &= (-5.745,0.436)\,, \end{aligned}\right\} \tag{5.1.5}$$

where the horizontal length scale is $L = 112$ km, and the depth scale is $H_0 = 200$ m. With these values for the shelf parameters and the resonant triad substituted into (2.2.16) and (2.2.21) we obtain the following numerical values for the interaction coefficents

$$\left.\begin{aligned} K_1 &= -8.757\,, \\ K_2 &= -2.054\,, \\ K_3 &= 4.613\,. \end{aligned}\right\} \tag{5.1.6}$$

The nondimensional friction parameter r, is given by (2.1.5) with the horizontal velocity field given by $V = 10^{-1}$m s^{-1}. The dimensional friction parameter r^*, is defined as the product of the Coriolis parameter with the square root of the vertical Ekman number E_V (Pedlosky 1987), which in turn, is given by

$$E_V = \frac{2A_V}{fH_0^2}\,. \tag{5.1.8}$$

Estimates for A_V range from 10^{-4}ms^{-1} to 10^{-1}ms^{-1} (Pedlosky 1987, section 4.2). Using the upper estimates for A_V we obtain the following damping coefficents

$$\left.\begin{aligned} r_1 &= 6.904\,, \\ r_2 &= 53.42\,, \\ r_3 &= 36.9\,, \end{aligned}\right\} \tag{5.1.9}$$

while using the lower estimates for A_V we obtain

$$\left.\begin{aligned} r_1 &= 0.006904\,, \\ r_2 &= 0.05342\,, \\ r_3 &= 0.0369\,. \end{aligned}\right\} \tag{5.1.9}$$

Since the damping coefficients are mutually different, analytical solutions to (5.1.3) and (5.1.4) can not be obtained. We note that if $\theta = \pi/2$ or $3\pi/2$ (the case where energy exchange is maximized), (5.1.4) is satisfied for all T, and (5.1.3) can be numerically integrated. The numerical solutions are plotted in figures 3 and 4, for the upper and lower estimates of r. While resonant interactions are a possible mechanism for energy exchange when Ekman friction is weak, we can see that strong bottom friction causes an e-folding time which is much smaller than the period of oscillation of the amplitudes.

To obtain numerical values for the period of oscillation of the amplitudes, we neglect friction in (5.1.3) so that the equations can be solved analytically. Setting $r_j = 0$, $(j = 1,2,3)$ and taking $\theta = \pi/2$, (5.1.3) reduces to

$$\left.\begin{aligned} \frac{da_1}{dT} &= -K_1 a_2 a_3\,, \\ \frac{da_2}{dT} &= -K_2 a_3 a_1\,, \\ \frac{da_3}{dT} &= -K_3 a_1 a_2\,. \end{aligned}\right\} \tag{5.1.10}$$

As shown by Hsieh and Mysak (1980) the solutions of (5.1.10) are the Jacobi elliptic functions. Without loss of generality we can choose $T = 0$ to be the

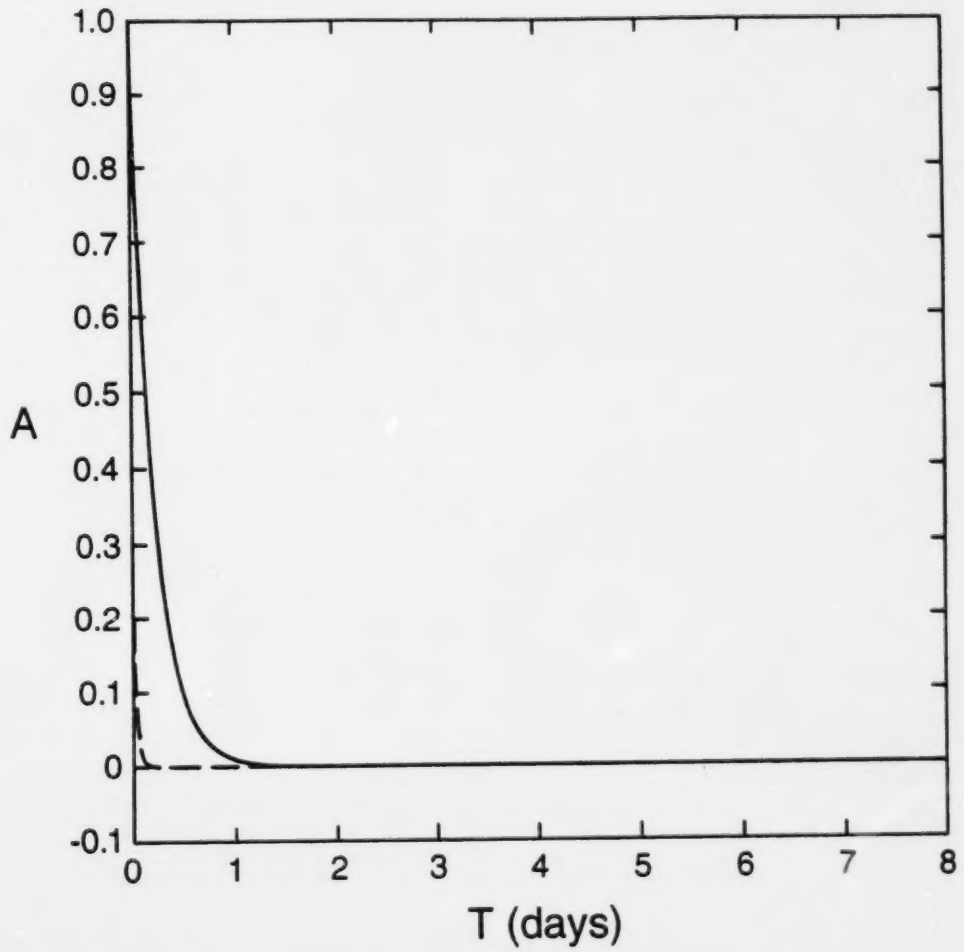


Fig. 3. Amplitude versus time for the upper estimate of the Ekman number, i.e., $r = 25.0$. Wave 1, wave 2 and wave 3 are denoted by a thick curve, a dashed curve and a dotted curve respectively.

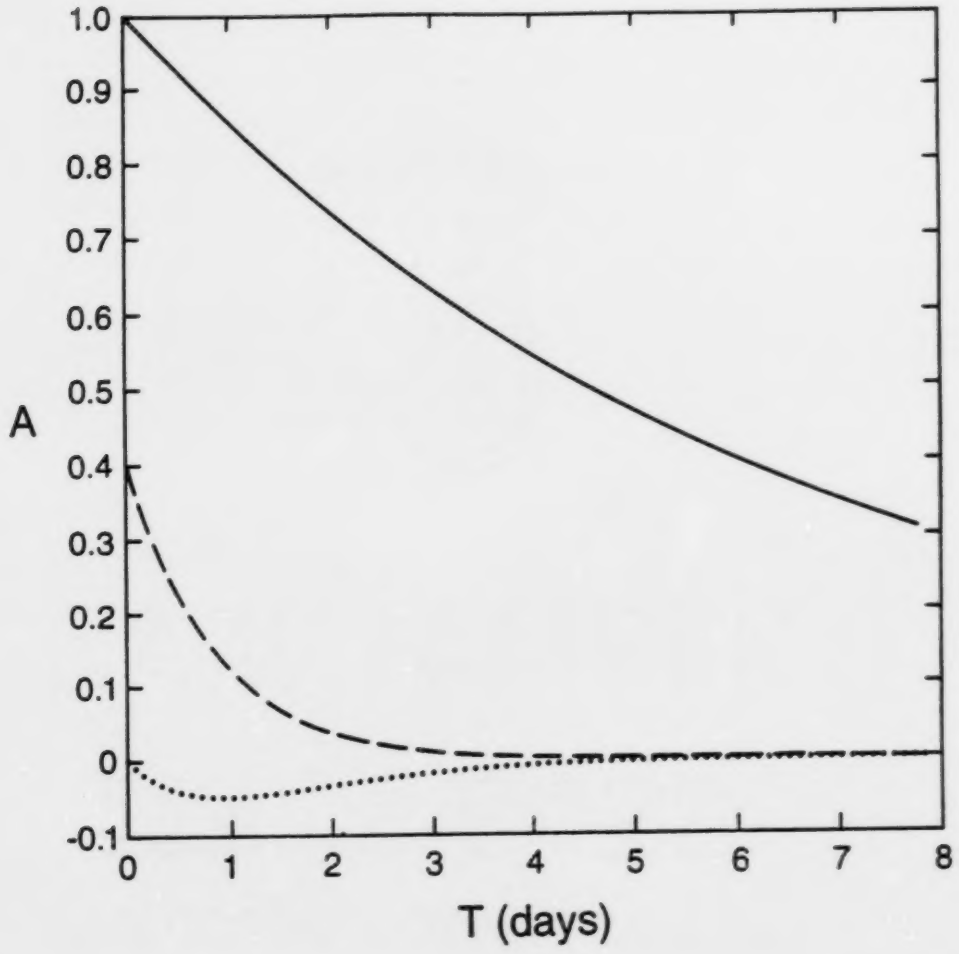


Fig 4. Amplitude versus time for the lower estimate of the Ekman number, i.e., $r = 0.8$. Wave 1, wave 2 and wave 3 are denoted by a thick curve, dashed curve and a dotted curve respectively.

instant when $a_1 > 0$, $a_2 > 0$ and $a_3 = 0$. With this choice of initial conditions ($a_1(0) \equiv a_{10} > 0$, $a_2(0) \equiv a_{20} > 0$, $a_{30} \equiv 0$), the solutions are

$$\left.\begin{aligned} a_1(T) &= a_{10} dn(\sigma T/M)\,, \\ a_2(T) &= a_{20} cn(\sigma T/M)\,, \\ a_3(T) &= a_{10}(-K_3/K_2)^{1/2} sn(\sigma T/M)\,, \end{aligned}\right\} \tag{5.1.11}$$

where

$$\sigma \equiv a_{10}(-K_2 K_3)^{1/2}\,, \tag{5.1.12,}$$

$$M \equiv K_1 a_{20}^2/(K_2 a_{10}^2)\,, \tag{5.1.13}$$

and dn, cn, sn are the Jacobi elliptic functions, Abramowitz and Stegun (1965, Chap. 16). Usually the elliptic functions are restricted to $0 \leq M \leq 1$. For $M > 1$, the following transformations can be used:

$$\left.\begin{aligned} dn(\sigma T/M) &= cn(M^{1/2}\sigma T/M^{-1})\,, \\ cn(\sigma T/M) &= dn(M^{1/2}\sigma T/M^{-1})\,, \\ sn(\sigma T/M) &= M^{1/2} sn(M^{1/2}\sigma T/M^{-1})\,. \end{aligned}\right\} \tag{5.1.14}$$

Without loss of generality we assume $0 \leq M \leq 1$. Then the period of energy transfer is given by the period T_d, of the elliptic function $dn(\sigma T/M)$. From Abramowitz and Stegun (1965 chaps. 16 and 17),

$$T_d = 2K(M)/\sigma = 2K(M) a_{10}^{-1}(-K_2 K_2)^{-1/2}\,, \tag{5.1.15}$$

where $K(M)$ is a complete elliptic integral of the fist kind. Using the same parameters and initial conditions ($a_{10} = 1, a_{20} = 0.4, a_{30} = 0$), used in the numerical solutions, we find $T_d \approx 1.33187$. Remembering that $T = \varepsilon t$, where ε is the Rossby number ($\varepsilon = 0.0089$), and that t is nondimensional with respect to $1/f$, ($f = 10^{-4}\mathrm{s}^{-1}$), T_d is converted back to dimensional units on division by εf, yielding $T_d \approx 1.491694 \times 10^6$ s or 17 days. Hence, for the given initial

conditions the time scale of energy transfer ($\sim 1/2T_d$) is 9 days. The e-folding time due to Ekman friction range from 0.5 days to 16.4 days. For the lower estimate of friction, the e-folding time is longer than the period of energy exchange so that some energy can be transfered to the other members of the triad before the waves are damped out. On the other hand, for the upper estimate of friction, the e-folding time is much longer than the period of energy exchange, and resonant interactions can not be expected to provide a mechanism for energy exchange between the three wave components. In fact for the upper estimate of friction our perturbation expansion may not be valid and friction may need to be included at first order.

Because the range of estimates in the Ekman number is so large, the theory developed in this thesis is inconclusive as to whether resonant interactions are a viable mechanism for resonant energy exchange in the presence of bottom friction. It is quite likely that the magnitude of the Ekman number varies with time and location, and this fact may explain in part why resonant interactions are not always present on all continental shelves.

Chapter 6

The Steady State Problem

6.1 Steady State Solution With Alongshore Topographic Variation

In this Chapter we focus on the effect of alongshore topographic variation on a resonant triad of waves. We neglect bottom friction and look at the steady-state problem in the presence of a piece-wise constant topographic configuration given by a Heavyside step function of the form

$$h_1(x,Y) = \begin{cases} H\,, & Y < Y_s \\ 0\,, & Y > Y_s \end{cases} \tag{6.1.1}$$

where the "jump" parameter H and the "shift" parameter Y_s are assumed constant. Exact nonlinear solutions can be obtained for the problem described above. The solution presented here forms the basic "building-block" in our subsequent work. In the next section we will use the solution obtained here to construct steady solutions over a ridge which will be modelled as piece-wise constant functions; see Figure 5.

A change in alongshore topography of the form (6.1.1) is an $O(\varepsilon)$ increase or reduction in water depth. In the $O(1)$ linear problem (2.1.16-2.1.18), this change in water depth results on an $O(\varepsilon)$ correction in the dispersion relationship. For example, if we use the exponential cross-shelf profile described in Chapter 4, a change in alongshore topography of the form (6.1.1) results on an $O(\varepsilon)$ correction in the parameter b appearing in (4.1.1). This correction in b results in an $O(\varepsilon)$ correction in the dispersion relationhsip (4.1.6). Clearly, if a set of triad wavenumbers and frequencies corresponds to a maximally interacting configuration at one point along the shelf, it will, in general, no longer be maximally interacting at a further point along the shelf if the water depth has changed in the alongshore direction.

If we use the exponential cross-shelf profile described in Chapter 4, substitution of (6.1.1) into (4.2.8) yields the expressions for the $\mu_j(Y)$'s ($j = 1,2,3$) defined in (2.2.25), i.e.,

$$\mu_j(Y) = \begin{cases} H\mu_{j0}\,, & Y < Y_s \\ 0\,, & Y > Y_s \end{cases} \tag{6.1.2}$$

where μ_{j0} is defined in (4.2.9). Substitution of (6.1.2) in (2.2.26) yields the following expression for $\sigma_j(Y)$, i.e.,

$$\sigma_j(Y) = \begin{cases} \mu_{j0}H(Y - Y_s)\,, & Y < Y_s \\ 0\,, & Y > Y_s \end{cases} \tag{6.1.3}$$

from which we obtain, after substitution into (2.2.29), the following expression for $\sigma_0(Y)$,

$$\sigma_0(Y) = \sigma H(Y - Y_s), \tag{6.1.4}$$

where

$$\sigma = \mu_{10}/c_{g1} + \mu_{20}/c_{g2} + \mu_{30}/c_{g3}.$$

Setting time-derivatives equal to zero and neglecting friction in (2.2.28) yields, after substitution of (6.1.3)

$$\begin{aligned} \alpha_{1\xi} &= -iK_{10}\alpha_2^*\alpha_3^*\exp[-i\sigma_0(\xi)] \\ \alpha_{2\xi} &= -iK_{20}\alpha_3^*\alpha_1^*\exp[-i\sigma_0(\xi)] \\ \alpha_{3\xi} &= -iK_{30}\alpha_1^*\alpha_2^*\exp[-i\sigma_0(\xi)] \end{aligned} \tag{6.1.5}$$

where $K_{j0} = K_j/c_{gj}$ for $j = 1,2,3$, and $\xi = Y - Y_s$. Cree and Swaters (1990) studied a similar system of equations derived in the context of Rossby wave interactions in the presence of linear topographic forcing. Wieland and Wilhelmsson (1977) and Craik (1985) have shown that if all the K_{j0}'s are of one sign, (6.1.5) will lead to explosively unstable solutions for the amplitudes. We know from Table 3, that even though the K_j's never all have the same sign (because of energy conservation), it is possible that the K_{j0}'s will all have the same sign if one of the c_{gj}'s has a different sign from the other two. Since our system

converves energy, it is unphysical to have solutions where the amplitudes become infinite. That our model seems to lead to explosively unstable solutions despite the fact that energy is conserved can be explained as follows. We note that the explosive instability only occures for triads where one of the c_{gj}'s has a different sign than the other two. This means that as time increases the wave envelopes rapidly separate, so that the waves can interact for only a short period of time. Triads that interact only briefly can not be expected to have steady state solutions. In fact steady state solutions for triads that lead to explosive instability would break our initial scaling demands and the theory would break down. It is important to remember that in the following work, the steady state solutions we will present apply only to resonant triads that have all the group velocities of the same sign.

The solution to (6.1.5) can be found in the form (Weiland and Wilhelmsson, 1977; Craik 1985)

$$\alpha_j = b_j(\xi)\exp[i\Phi_j(\xi)]\,, \tag{6.1.6}$$

for $j = 1, 2$ and 3. The $b_j(\xi)$ functions correspond to the envelope amplitudes and the $\Phi_j(\xi)$ correspond to envelope phase functions. Substitution of (6.1.6) in (6.1.5) leads to, after separating real and imaginary parts,

$$b_{1\xi} = K_{10} b_2 b_3 \sin(\Phi) \tag{6.1.7a}$$

$$b_{2\xi} = \mp K_{20} b_3 b_1 \sin(\Phi) \tag{6.1.7b}$$

$$b_{3\xi} = K_{30} b_1 b_2 \sin(\Phi) \tag{6.1.7c}$$

$$\Phi_{1\xi} = -K_{10}(b_2 b_3 / b_1)\cos(\Phi) \tag{6.1.8a}$$

$$\Phi_{2\xi} = -K_{20}(b_3 b_1 / b_2)\cos(\Phi) \tag{6.1.8b}$$

$$\Phi_{3\xi} = -K_{30}(b_1 b_2 / b_3)\cos(\Phi) \tag{6.1.8c}$$

where

$$\Phi(\xi) = -\Phi_1(\xi) - \Phi_2(\xi) - \Phi_3(\xi) - \sigma_0(\xi) \tag{6.1.9}$$

Equations (6.1.8) and (6.1.9) can be combined to yield

$$\Phi_\xi = -\sigma H + b_1 b_2 b_3 (K_{10}/b_1^2 + K_{20}/b_2^2 + K_{30}/b_3^2)\cos(\Phi) \tag{6.1.10}$$

We note that the solution for the envelope amplitudes (i.e. b_j's) is obtained from the four equations (6.1.7) and (6.1.10), while the envelope phases (i.e., Φ_j's) will be obtained from (6.1.9) and any two of (6.1.8).

It is possible to see qualitatively how topography acts to dephase the wave-wave interactions based on (6.1.7), (6.1.8) and (6.1.10). It follows from (6.1.8) that maximum energy exchange occurs for $\sin(\Phi) = \pm 1$ or equivalently $\Phi = \pm(2n+1)\pi/2$ [consequently $\cos(\Phi) = 0$], where n is a nonengative integer. We may assume, without loss of generality, that initialy $\Phi_1 = \pi/2$, $\Phi_2 = \Phi_3 = 0$. In the absence of any alongshore topographic variation (i.e., $H = 0$), it will follow from (6.1.10) that Φ remains equal to $-\pi/2$ thereafter and thus from (6.1.8) it follows that $\Phi_1 = \pi/2$, $\Phi_2 = \Phi_3 = 0$ for all ξ. In addition, from (6.6) it follows that the energy exchange will remain maximized because $\sin\Phi = -1$. On the other hand if $\sigma_0 \neq 0$ in (6.1.10), then $\Phi_\xi \neq 0$ initially and the change in water depth will act to induce a nonzero Φ and thus, on account of (6.1.8), the envelope phases will have magnitudes which will diverge from their initial values. Also, as Φ moves away from $-\pi/2$, the interaction coefficients in (6.1.7) will reduce in magnitude and the energy exchange will no longer be maximized. Following Cree and Swaters (1991), we will call the evolution of the Φ_j's the "dephasing" aspect of the topographically modified interaction, and the role of the b_j's in (6.1.7) the "amplitude modulation" aspect of the interaction.

To obtain the solution to (6.1.7) and (6.1.10) we proceed as follows. First, we introduce the normalizations

$$b_1 = |K_{20}K_{30}|^{-1/2}\widehat{b}_1, \tag{6.1.11a}$$

$$b_2 = |K_{30}K_{10}|^{-1/2}\widehat{b}_2, \tag{6.1.11b}$$

$$b_3 = |K_{10}K_{20}|^{-1/2}\widehat{b}_3. \tag{6.1.11c}$$

Substitution of (6.1.11) into (6.1.7) and (6.1.10) yields

$$\widehat{b}_{1\xi} = s_1 \widehat{b}_2 \widehat{b}_3 \sin(\Phi), \tag{6.1.12a}$$

$$\widehat{b}_{2\xi} = s_2 \widehat{b}_3 \widehat{b}_1 \sin(\Phi), \tag{6.1.12b}$$

$$\widehat{b}_{3\xi} = s_3 \widehat{b}_1 \widehat{b}_2 \sin(\Phi), \tag{6.1.12c}$$

$$\Phi_\xi = -\sigma H + \widehat{b}_1 \widehat{b}_2 \widehat{b}_3 (s_1/\widehat{b}_1^2 + s_2/\widehat{b}_2^2 + s_3/\widehat{b}_3^2) \cos(\Phi) \tag{6.1.13}$$

where $s_j = \mathrm{sgn}(K_{j0})$ with $j = 1, 2, 3$. It follows immediately from energy conservation, that two of the K_{j0} are of one sign and the other differs in sign, and because of this, two of the s_j are of one sign while the other differs in sign. Without loss of generality we can choose $s_2 = s_3 = 1$ and $s_1 = -1$.

The Manley-Rowe realtions are satisfied by (6.1.12). To see this, we form the products $-\widehat{b}_1 \times$(6.1.12a), $-\widehat{b}_2 \times$(6.1.12b), $-\widehat{b}_3 \times$(6.1.12c), and integrate with respect to ξ over the interval (ξ, ξ_0), to obtain

$$[\widehat{b}_1^2(\xi) - \widehat{b}_1^2(\xi_0)] = -[\widehat{b}_2^2(\xi) - \widehat{b}_2^2(\xi_0)] = -[\widehat{b}_3^2(\xi) - \widehat{b}_3^2(\xi_0)]. \tag{6.1.14}$$

It will be convenient for our future work to introduce the auxiliary dependent variable

$$\widehat{y}(\xi) = -[\widehat{b}_1^2(\xi) - \widehat{b}_1^2(\xi_0)], \tag{6.1.15}$$

which on account of (6.1.14) also satisfies $\widehat{y}(\xi) = \widehat{b}_j^2(\xi) - \widehat{b}_j^2(\xi_0)$ for $j = 2, 3$.

Another auiliary variable that will be convenient for our subsequent work is the constant of the motion given by

$$\Gamma = \widehat{b}_1 \widehat{b}_2 \widehat{b}_3 \cos(\Phi) + \sigma H \widehat{b}_1^2/2. \tag{6.1.16}$$

To show that Γ is a constant of the motion it is sufficient to show $\Gamma_\xi = 0$. This can be done by taking the derivative of the left and right-hand sides of (6.1.16) with respect to ξ and eliminate $\widehat{b}_{j\xi}$ and Φ_ξ using (6.1.12) and (6.1.13).

We are now in a position to derive an ordinary differential equation for $\widehat{y}(\xi)$. The solution $\widehat{y}(\xi)$, will be given in terms of elliptic functions. With $\widehat{y}(\xi)$ known, the Manley-Rowe relations will determine the $\widehat{b}_j(\xi)$ for $j = 1, 2$, and 3.

If we multiply (6.11b) by $\widehat{b}_2(\xi)$, it follows that

$$(\widehat{b}_2^2)_\xi = \pm 2\{\widehat{b}_1^2\widehat{b}_2^2\widehat{b}_3^2[1 - \cos^2(\Phi)]\}^{1/2}. \tag{6.1.17}$$

Now if (6.1.15) and the Manley-Rowe relations are used to eliminate the $\widehat{b}_j^2$ in (6.1.17), and (6.1.16) is used to eliminate $\cos^2(\Phi)$ in (6.1.17), it follows that

$$\frac{1}{2}(\frac{\partial \widehat{y}}{\partial \xi})^2 + p(\widehat{y}) = 0 \tag{6.1.18}$$

where

$$p(\widehat{y}) \equiv -2\{[\widehat{b}_1^2(\xi_0) - \widehat{y}][\widehat{b}_2^2(\xi_0) + \widehat{y}][\widehat{b}_3^2(\xi_0) + \widehat{y}] - [\Gamma - \frac{\sigma H}{2}(\widehat{b}_1^2(\xi_0) - \widehat{y})]^2\}. \tag{6.1.19}$$

The polynomial $p(\widehat{y})$ is cubic in the dependent variable $\widehat{y}(\xi)$. Equation (6.1.18) can be seen as describing the motion of a particle in a nonliear potential $p(\widehat{y})$, where $\frac{1}{2}(\partial\widehat{y}/\partial\xi)^2$ represents the kinetic energy. In this formallism the coordinate $\widehat{y}$, giving the position of the particle, is proportional to the deviation of the squared values of the wave amplitudes from their initial values. We also note that only the range of $\widehat{y}$ for which $p(\widehat{y}) \leq 0$ has a physical significance, and $\partial\widehat{y}/\partial\xi = 0$ when $p(\widehat{y}) = 0$. It is possible to show (Weiland and Wilhelmsson, 1977; Craik 1985) that because not all the s_j coefficients in (3.1.8) have the same sign (which followed from energy conservation), $p(\widehat{y})$ will have three real roots, $\widehat{y}_1 > \widehat{y}_2 > \widehat{y}_3$. The solution $\widehat{y}(\xi)$ will, in general, oscillate between the two largest roots of $p(\widehat{y})$. We can take advantage of this fact to perform a change of variables in (6.1.18). Since $\widehat{y}_2 \leq \widehat{y}(\xi) \leq \widehat{y}_1$, we can define a new dependent varialbe as follows:

$$\theta(\xi) \equiv \sin^{-1}(\sqrt{\frac{\widehat{y}_1 - \widehat{y}(\xi)}{\widehat{y}_1 - \widehat{y}_2}}). \tag{6.1.20}$$

It is easy to see that the following three equations follow directly from (6.1.20):

$$\widehat{y} - \widehat{y}_1 = -(\widehat{y}_1 - \widehat{y}_2)\sin^2(\theta), \tag{6.1.21}$$

$$\widehat{y} - \widehat{y}_2 = -(\widehat{y}_1 - \widehat{y}_2) - (\widehat{y}_1 - \widehat{y}_2)\sin^2(\theta), \tag{6.1.22}$$

$$\widehat{y} - \widehat{y}_3 = -(\widehat{y}_1 - \widehat{y}_3) - (\widehat{y}_1 - \widehat{y}_2)\sin^2(\theta). \tag{6.1.23}$$

Differentiating (6.1.21) with respect to ξ yields

$$\widehat{y}_\xi = -2(\widehat{y}_1 - \widehat{y}_2)\sin(\theta)\cos(\theta)\theta_\xi. \tag{6.1.24}$$

By first writting (6.1.18) as follows

$$\frac{1}{2}(\widehat{y}_\xi)^2 = -2(\widehat{y} - \widehat{y}_1)(\widehat{y} - \widehat{y}_2)(\widehat{y} - \widehat{y}_3), \tag{6.1.25}$$

Then substituting (6.1.21), (6.1.22), (6.1.23) and (6.1.24) into (6.1.25) we obtain, after simplification, an equation in θ,

$$(\theta_\xi)^2 = (\widehat{y}_1 - \widehat{y}_3)(1 - \frac{\widehat{y}_1 - \widehat{y}_2}{\widehat{y}_1 - \widehat{y}_3}\sin^2\theta). \tag{6.1.26}$$

For convenience we rewrite this last equation as follows

$$\theta_\xi = \pm\sqrt{\widehat{y}_1 - \widehat{y}_3}(1 - \sin^2(\alpha)\sin^2\theta)^{1/2}, \tag{6.1.27}$$

where

$$\sin(\alpha) \equiv \sqrt{\frac{\widehat{y}_1 - \widehat{y}_2}{\widehat{y}_1 - \widehat{y}_3}}. \tag{6.1.28}$$

Separating variables and integrating yields

$$\pm\sqrt{\widehat{y}_1 - \widehat{y}_3}(\xi - \xi_0) = \int_{\theta(\xi_0)}^{\theta(\xi)} \frac{d\theta}{\sqrt{1 - \sin^2\alpha\sin^2\theta}} \tag{6.1.29}$$

where ξ_0 was chosen so that $\widehat{y}(\xi_0) = 0$. The integral on the right hand side of (6.1.29) can be seen as the difference of elliptic integrals of the first kind

(Abramowitz and Stegun, 1964 Chap 17). Thus, in the notation of Abramowitz and Stegun, (6.1.29) becomes

$$\pm\sqrt{\widehat{y}_1 - \widehat{y}_3}(\xi - \xi_0) = F[\theta(\xi)\backslash\alpha] - F[\theta(\xi_0)\backslash\alpha] \tag{6.1.30}$$

By definition

$$sn(F[\theta(\xi)\backslash\alpha]) = \sin(\theta(\xi)), \tag{6.1.31}$$

so that (6.1.30) becomes

$$\sin(\theta(\xi)) = sn(\pm\sqrt{\widehat{y}_1 - \widehat{y}_3}(\xi - \xi_0) + F[\theta(\xi_0)\backslash\alpha]). \tag{6.1.32}$$

Squaring (6.1.32) and eliminating $\sin^2(\theta)$ by using (6.1.21) yields

$$\widehat{y} = (\widehat{y}_2 - \widehat{y}_1)sn^2(\pm\sqrt{\widehat{y}_1 - \widehat{y}_3}(\xi - \xi_0) + \widehat{\theta}|\widehat{k}) + \widehat{y}_1, \tag{6.1.33}$$

where

$$\widehat{k} \equiv \sin\alpha = \sqrt{\frac{\widehat{y}_1 - \widehat{y}_2}{\widehat{y}_1 - \widehat{y}_3}}, \tag{6.1.34}$$

is the modulus, and

$$\widehat{\theta} \equiv F(\theta(\xi_0)\backslash\alpha) = sn^{-1}(\sqrt{\frac{\widehat{y}_1}{\widehat{y}_1 - \widehat{y}_2}}). \tag{6.1.35}$$

The scaled wavelength of the interaction is given by

$$\lambda_I = 2\int_0^{2\pi}(1 - \widehat{k}\sin^2\xi)^{-1/2}d\xi \tag{6.1.36}$$

With the solution for $\widehat{y}(\xi)$ given by (6.1.32) the solution for the amplitudes can be written in the form

$$\widehat{b}_1(\xi) = \pm[\widehat{b}_1^2(\xi_0) - \widehat{y}(\xi)]^{1/2}, \tag{6.1.37a}$$

$$\widehat{b}_2(\xi) = \pm[\widehat{b}_2^2(\xi_0) - \widehat{y}(\xi)]^{1/2}, \tag{6.1.37b}$$

$$\widehat{b}_3(\xi) = \pm[\widehat{b}_3^2(\xi_0) - \widehat{y}(\xi)]^{1/2}. \tag{6.1.37c}$$

The solution for $\Phi(\xi)$ may be obtained by eliminating $\hat{b}_1\hat{b}_2\hat{b}_3\cos(\Phi)$ in (6.1.13) using (6.1.16) and integrating to yield

$$\Phi(\xi) = \Phi(\xi_0) - \sigma H(\xi - \xi_0) + \int_{\xi_0}^{\xi} [\Gamma - \sigma H\hat{b}_1^2(\eta)/2][-\hat{b}_1^{-2}(\eta) + \hat{b}_2^{-2}(\eta) + \hat{b}_3^{-2}(\eta)]d\eta \tag{6.1.38}$$

Similarly, the individual envelope phases may be obtained from (6.1.8), to yield

$$\Phi_1(\xi) = \Phi_1(\xi_0) + \int_{\xi_0}^{\xi} \hat{b}_2(\eta)\hat{b}_3(\eta)\cos[\Phi(\eta)]/\hat{b}_1(\eta)d\eta\ , \tag{6.1.39a}$$

$$\Phi_2(\xi) = \Phi_2(\xi_0) + \int_{\xi_0}^{\xi} \hat{b}_1(\eta)\hat{b}_3(\eta)\cos[\Phi(\eta)]/\hat{b}_2(\eta)d\eta\ , \tag{6.1.39b}$$

$$\Phi_3(\xi) = \Phi_3(\xi_0) + \int_{\xi_0}^{\xi} \hat{b}_1(\eta)\hat{b}_2(\eta)\cos[\Phi(\eta)]/\hat{b}_3(\eta)d\eta\ , \tag{6.1.39c}$$

For computational purposes, we can eliminate $\cos[\Phi(\eta)]$ in (6.1.39) by using (6.1.16) to obtain

$$\Phi_1(\xi) = \Phi_1(\xi_0) + \int_{\xi_0}^{\xi} \frac{-(2\Gamma - \sigma H\hat{b}_1^2(\eta))}{2\hat{b}_1^2(\eta)}d\eta, \tag{6.1.40a}$$

$$\Phi_2(\xi) = \Phi_2(\xi_0) + \int_{\xi_0}^{\xi} \frac{-(2\Gamma - \sigma H\hat{b}_1^2(\eta))}{2\hat{b}_2^2(\eta)}d\eta, \tag{6.1.40b}$$

$$\Phi_3(\xi) = \Phi_3(\xi_0) + \int_{\xi_0}^{\xi} \frac{-(2\Gamma - \sigma H\hat{b}_1^2(\eta))}{2\hat{b}_3^2(\eta)}d\eta\ ., \tag{6.1.40c}$$

6.2. Triad Interactions Over a Simple Piece-wise Constant Topographic Configuration

The solution presented in Section 6.1 can be used as a building-block to obtain continuous solutions for simple topographic configurations which are constructed as piece-wise constant sections. We will look at a topographic

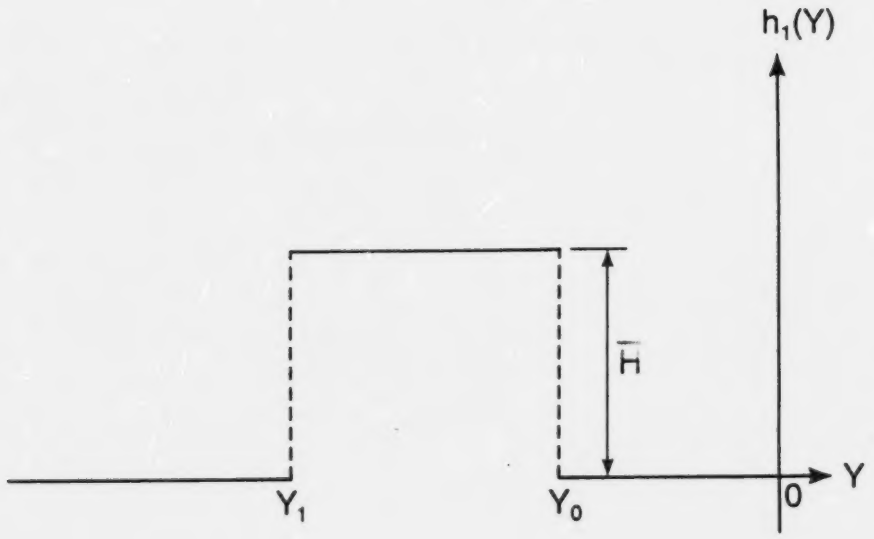


Fig. 5. Alongshore topography profile for a ridge modelled as a piece-wise constant configuration.

configuration that is modelling a ridge as illustrated in Figure 5. The topography shown in Figure 5 corresponds to

$$h_1(x,Y) = \begin{cases} 0\,, & Y_0 < Y < 0 \\ H\,, & Y_1 < Y < Y_0 \\ 0\,, & -\infty < Y < Y_1\,. \end{cases} \tag{6.2.1}$$

For our steady-state solution to be valid we need one of the K_{j0}'s to be of a different sign than the other two. Remembering that $K_{j0} = K_j/c_{gj}$, we see from Table 3 that the K_{j0}'s will differ in sign when all the c_{gj}'s are negative, and because of this the energy will propagate in the negative Y direction. Hence we will impose our initial conditions to the right of the topographic feature, i.e., at $Y = 0$. In each individual Y-interval; $Y < Y_s$ and $Y_s > Y$, the solution will be of the form derived in section (6.2) for an appropriately determined shift parameter Y_s and jump parameter H in (6.1.1) in accordance with (6.2.1). Since, in general, the Y_s parameter will be different for the individual regions, the auxiliary independent variable ξ used in the solutions presented in section 6.2 will be different for the individual Y-intervals in (6.2.1). Nevertheless, the solutions for the envelope amplitude and phase functions can be made continuous at the Y-interval boundaries (i.e., Y_0 and Y_1) by choosing the initial values of the b_j's of the interval just entered, to be the final values of the amplitudes of the region just exited. One then recomputes the new roots $\widehat{y}_1$, $\widehat{y}_2$ and $\widehat{y}_3$ of the polynomial (6.1.19) with the new initial values of the b_j's. The parameter ξ_0 in (6.1.29) and (6.1.32) is then determined so the new $\widehat{y}$ satisfies $\widehat{y}(\xi_0) = 0$ at the Y-interval boundary point just passed. Also, at each Y-interval boundary point, one must determine which sign is appropriate for the argument of sn in (6.1.33). The correct sign is chosen so that when the right hand side of (6.1.12a) is positive at the boundary point then $\widehat{b}_1$ must be increasing to the left of the boundary point, and vice versa if the right hand side is negative. Forcing $\widehat{b}_1$ to be increasing or decreasing will fix the correct sign for sn's argument in (6.1.33).

In the numerical examples which we will describe, the members of the triad will all have negative group velocity and hence they will all propagate their energy with the coast to the right in the northern hemisphere. Consequently, we will now describe the appropriate parameter values to be used, assuming the topographic feature (6.2.1) is being traversed from right to left. For this situation we have

$$\left.\begin{aligned} Y_s &= 0, \\ \xi &= Y, \\ H &= 0, \\ \xi_0 &= 0, \end{aligned}\right\} \text{ in } Y_0 \le Y \le 0 \tag{6.2.2a}$$

$$\left.\begin{aligned} Y_s &= Y_0, \\ \xi &= Y - Y_0, \\ H &= \overline{H}, \\ \xi_0 &= 0, \end{aligned}\right\} \text{ in } Y_1 \le Y \le Y_0 \tag{6.2.2b}$$

$$\left.\begin{aligned} Y_s &= Y_1, \\ \xi &= Y - Y_1, \\ H &= 0, \\ \xi_0 &= 0\,. \end{aligned}\right\} \text{ in } Y_0 < Y_1 \tag{6.2.2c}$$

The wavenumbers and frequencies for the triad used in this example are given approximately by

$$\left.\begin{aligned} (k_1, \omega_1) &= (-2.592, 0.5292), \\ (k_2, \omega_2) &= (1.15, -0.3819), \\ (k_3, \omega_3) &= (1.442, -0.1473). \end{aligned}\right\} \tag{6.2.3}$$

The corresponding group velocities are given approximately by

$$\left.\begin{aligned} c_{g1} &= -0.01925, \\ c_{g2} &= -0.2163, \\ c_{g3} &= -0.08274. \end{aligned}\right\} \tag{6.2.4}$$

The interaction coefficients in (6.1.4) are given approximately by

$$\left.\begin{aligned} K_{10} &= -210.8, \\ K_{20} &= 22.03, \\ K_{30} &= 6.818. \end{aligned}\right\} \tag{6.2.5}$$

The topographic parameter σ in (6.1.4) is given approximately by

$$\sigma = 0.8206, \tag{6.2.6}$$

where

$$\left.\begin{aligned} \mu_{10} &= 0.7003, \\ \mu_{20} &= -0.70a03, \\ \mu_{30} &= 0.8687. \end{aligned}\right\} \tag{6.27}$$

We will, for this example set

$$\left.\begin{aligned} Y_0 &= -1, \\ Y_1 &= -3. \end{aligned}\right\} \tag{6.2.8}$$

The initial values of the envelope amplitudes b_j's, and the phase Φ_j's will be specified at the position $Y = 0$. Since we will assume that initially the resonant triad is maximally interacting, the initial envelope phases will be given by

$$\left.\begin{aligned} \Phi_1(0) &= \pi/2, \\ \Phi_2(0) &= 0, \\ \Phi_3(0) &= 0. \end{aligned}\right\} \tag{6.2.9}$$

The wavelength of the wave-wave interaction (i.e. the distance Y over which the energy exchange goes through one complete cycle) is determined by the roots of the cubic potential, which in turn depend in part on the wave initial amplitudes; see (6.1.35), (6.1.33) and (6.1.18). For the simulation described here, we have chosen the initial conditions

$$\left.\begin{aligned} b_1(0) &= 1.0\ , \\ b_2(0) &= 0.0029\ , \\ b_3(0) &= 0.0010\ , \end{aligned}\right\} \tag{6.2.10}$$

for the amplitudes given in (6.1.7). These initial values imply that the interaction nondimensional wavelength in the absence of alongshore topography is about one nondimensional Y-unit.

In Figure 6, we show the energy enchange cycle from $Y = 0.0$ to $Y = -5.0$ in the absence of topography. Wave 1, wave 2 and wave 3 are denoted by a thick curve, dashed curve and dotted curve respectively. We note in this figure that the energy flow among the members of the triad, in the absence of alongshore topography, pulsates with Y, and that the direction of energy flow is reversible. The wave with the interaction coefficient which has a sign opposite to the other two, in our case wave 1, will always vary in the opposite direction to the other two, and in this way the wave energy will oscillate between the members of the triad.

In Figure 7 we show how the energy exchange is modified for a "small" topographic jump parameter of $\leq H = 0.5$. In the initial region where the water depth has not yet changed, $-1.0 < Y < 0.0$, the energy exchange is unforced and is identical to that depicted in Figure 6. However, over the region where the water depth has changed $-3.0 < y < -1.0$, the energy partition has been modified; Figure 7A. While energy exchange still occurs, it does so in a less complete fashion in that the magnitude of the amplitude of wave 1 never decreases to zero while the maximum values of the amplitudes of waves 2 and 3 are reduced. Also, the effective wavelength of energy exchange over the topographic feature has been increased to about 1.2 units. Once the topographic feature has been traversed ($Y \leq -3.0$), the energy exchange nearly returns to its pre-topographic level, although the wavelength of interaction is greatly reduced.

In Figure 7b, $\Phi(Y)$ is plotted. Before the triad encounters the topographic feature, $\Phi(Y)$ remains constant with the value of $-\pi/2$, so that $\sin(\Phi) = -1$, and the energy exchange is maximal. Once the topographic feature is encountered, $\Phi(Y)$ begins to increase. This change in Φ is caused by the forcing due to the nonzero jump parameter ($H \neq 0$) in (6.1.34). The alongshore change in water depth causes the dephasing of the interaction and reduces the energy exchange.

After the topographic feature is passed Φ is no longer constant and the energy exchange remains permanently suppressed.

In Figure 8 we present a solution for a topographic configuration similar to the one presented in Figure 7, except that the length of the ridge has been reduced from 2.0 units to 1.72 units; ie $Y_1 = -2.72$. Although the solution for the regions prior to, and over the ridge is identical to the one shown in Figure 7, downstream from the ridge ($Y < Y_1$), the solution is dramatically different. In Figure 8, resonant interactions are negligible downstream from the topographic feature, while the resonant interactions returned almost completely to their pre-topographic levels in Figure 7. Changing the position of the shift parameter Y_1, in effect changes the initial values for the solution to the left of the topographic feature. We can see from Figures 7 and 8 that the strength of the resonant interaction downstream from a topographic feature is highly dependent on where the change in water depth occurs relative to the value of the phase variable.

The phase variable Φ is critical in understanding the permanent wavenumber mismatch induced by the alongshore topographic forcing. If we backtrack through transformations (6.1.5), (2.2.27) and (2.1.22), we can write the equations for the individual stream functions

$$\left.\begin{aligned}
\psi_1^{(0)} &= b_1(\varepsilon y)\phi(x)\exp[i(\theta_1 + \mu_{10}h_1(Y)/c_{g1} + \Phi_1(Y))],\\
\psi_2^{(0)} &= b_2(\varepsilon y)\phi(x)\exp[i(\theta_2 + \mu_{20}h_1(Y)/c_{g2} + \Phi_2(Y))],\\
\psi_3^{(0)} &= b_3(\varepsilon y)\phi(x)\exp[i(\theta_3 + \mu_{30}h_1(Y)/c_{g3} + \Phi_3(Y))],
\end{aligned}\right\} \tag{6.2.11}$$

where θ_j is the "fast" phase variable given by $k_j y - \omega_j t$. The total phase of the j^{th} wave in the triad can be written in the form

$$\psi_j = \theta_j - \mu_{j0}h_1(Y)/c_{gj} + \Phi_j(Y). \tag{6.2.12}$$

As in WKB theory we can think of the true wavenumber of the j^{th} wave as being a slowly varying function of y. Thus the alongshore wavenumber $\kappa(\varepsilon y)$ is given by

$$\kappa_j = \frac{\partial \psi_j}{\partial y} = k_j + \varepsilon[(\partial \Phi_j / \partial Y) + \mu_{j0} h_{1Y} / c_{gj}] \tag{6.2.13}$$

for $j = 1, 2$, and 3. If the sum $\kappa_1 + \kappa_2 + \kappa_3$ is formed, it follows upon using (6.1.13), and the resonance condition (2.2.11) that

$$\sum_{j=1}^{3} \kappa_j = -\varepsilon \Phi_Y \tag{6.2.14}$$

Consequently, if $\Phi_Y \neq 0$, the resonance condition on the complete wave number will no longer be satisfied. We can interpret the $\varepsilon \Phi_Y$ term in (6.2.14) as the alongshore wave number mismatch.

In Figures 10, 11 & 12 we present solutions over ridges of various length with jump parameter $H = 0.75$. The increase in the jump parameter further inhibits the exchange of energy over the ridge. The energy exchange seems to recover more completely, however, downstream from the ridge.

We have seen that a change in water depth can reduce the strength of the resonant interactions, and in fact, if the change in water depth is large enough the interactions may be completely eliminated. The resonant interactions are also permanently suppressed downstream from the topographic feature due to a permanent wavenumber mismatch induced by the topography. The strength of the wavenumber mismatch is not a monotonic function of the height of the topographic feature as might be expected, but in fact depends on the value of the magnitude of the wave amplitudes and the phase variable at the end point of the topographic feature. Thus the height of the topographic feature influences the strength of the wavenumber mismatch downstream only in as much as it changes the wavelength of the interaction over the topographic feature, so that the initial value of the phase variable will be different when the triad encounters the post-topographic region.

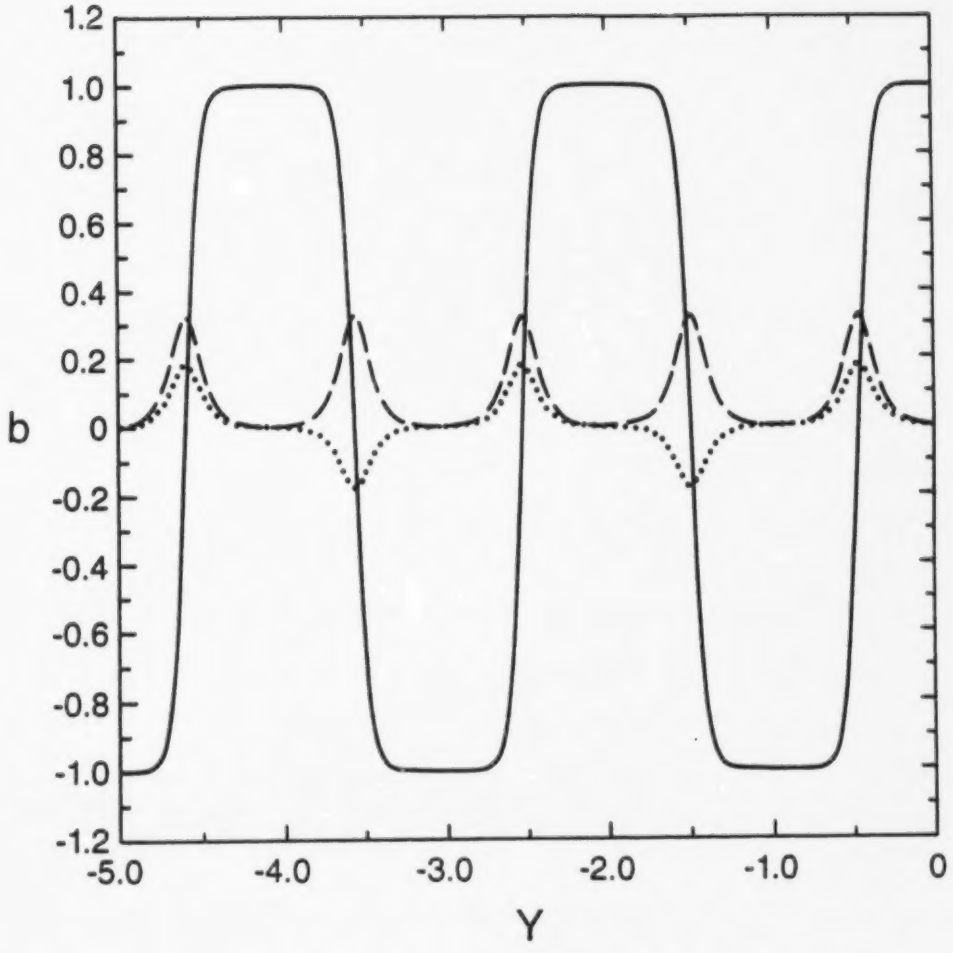


Fig. 6. Plot of b_j versus Y for initial amplitudes (6.2.10), initial phases (6.2.9), and topography given by (6.2.1) with $H = 0, 0$. Wave 1, wave 2 and wave 3 are denoted by a thick curve, dashed curve and a dotted curve respectively.

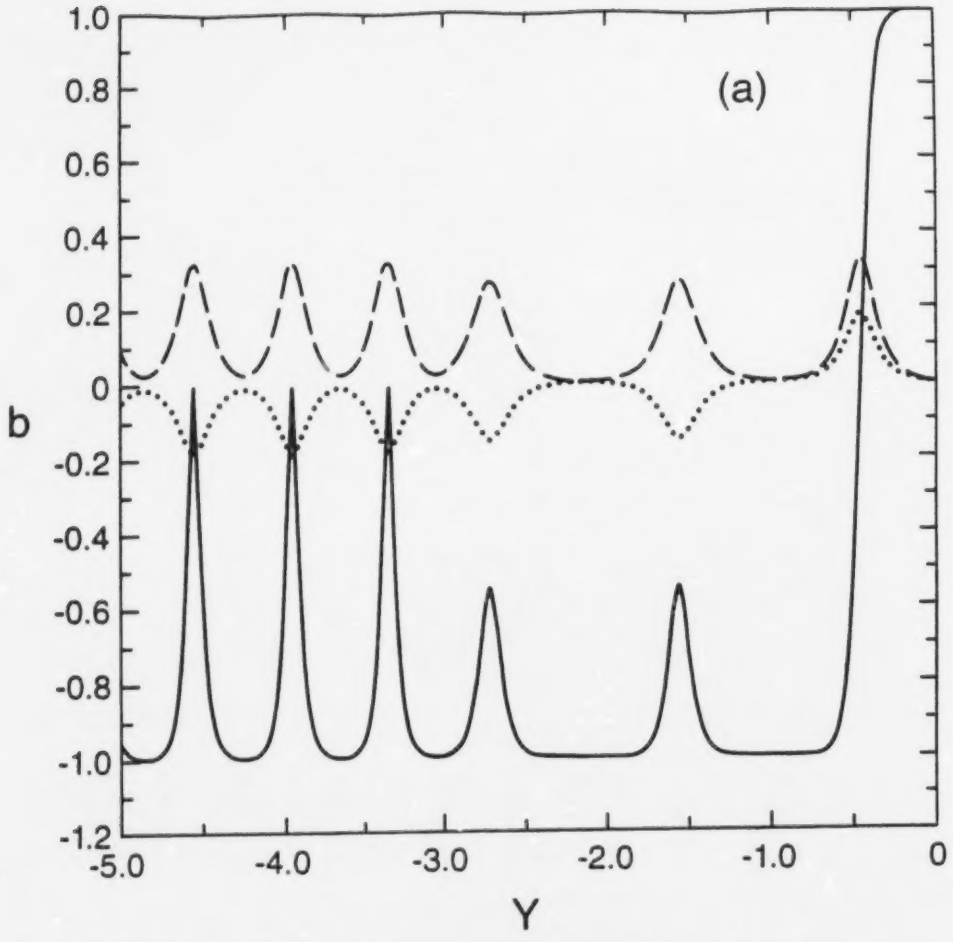


Fig. 7. Plots for initial Amplitudes (6.2.10), initial phases (6.2.9), and topography given in (6.2.1) with $H = 0.5$, $Y_0 = -1.0$ and $Y_1 = -3.0$.

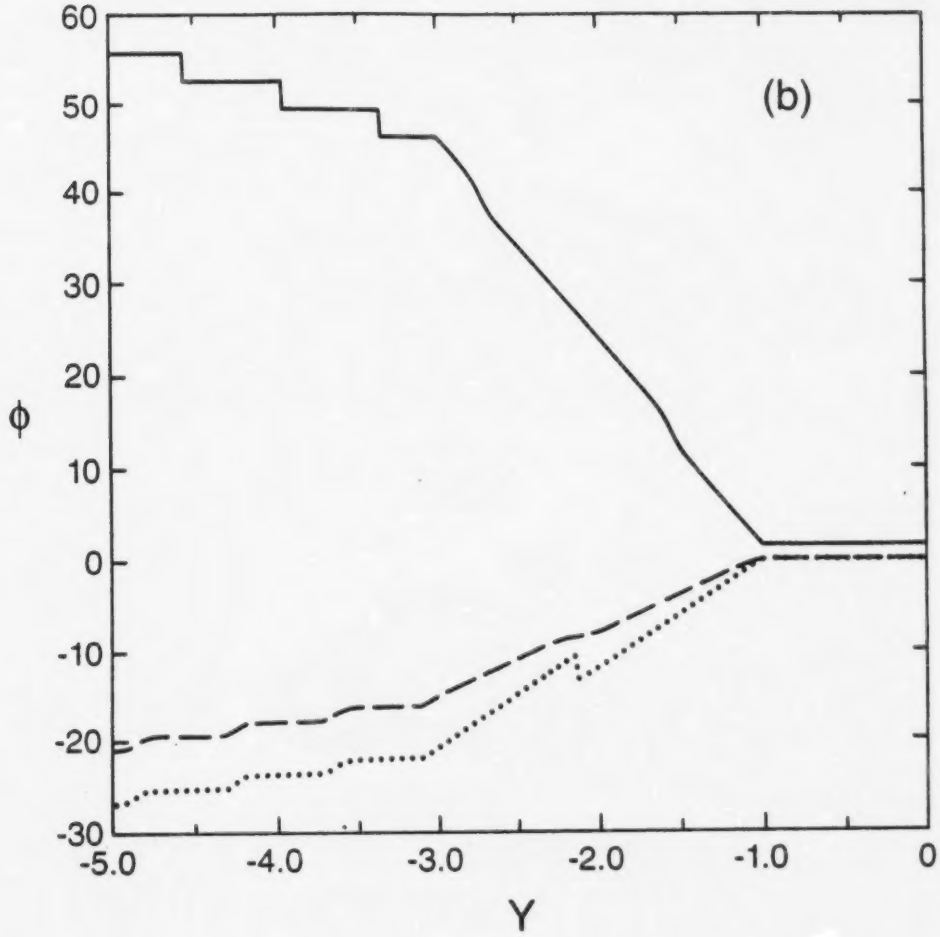
(A) b_j versus Y

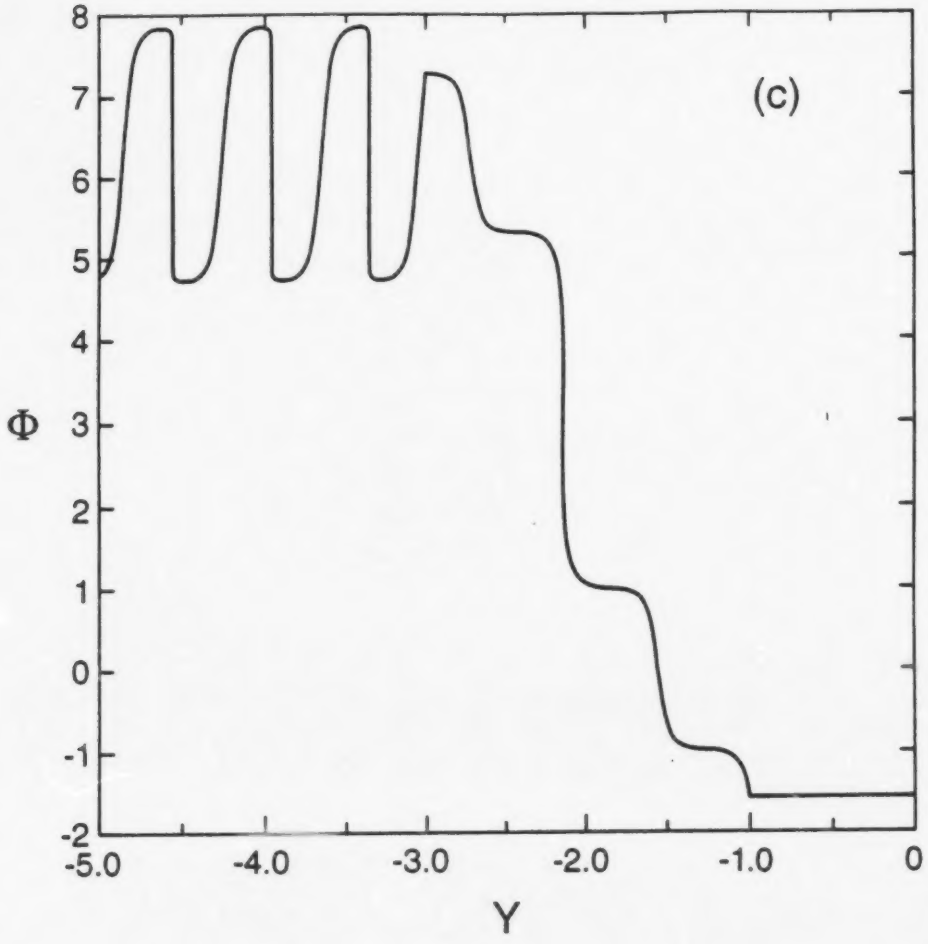
(B) Φ versus Y

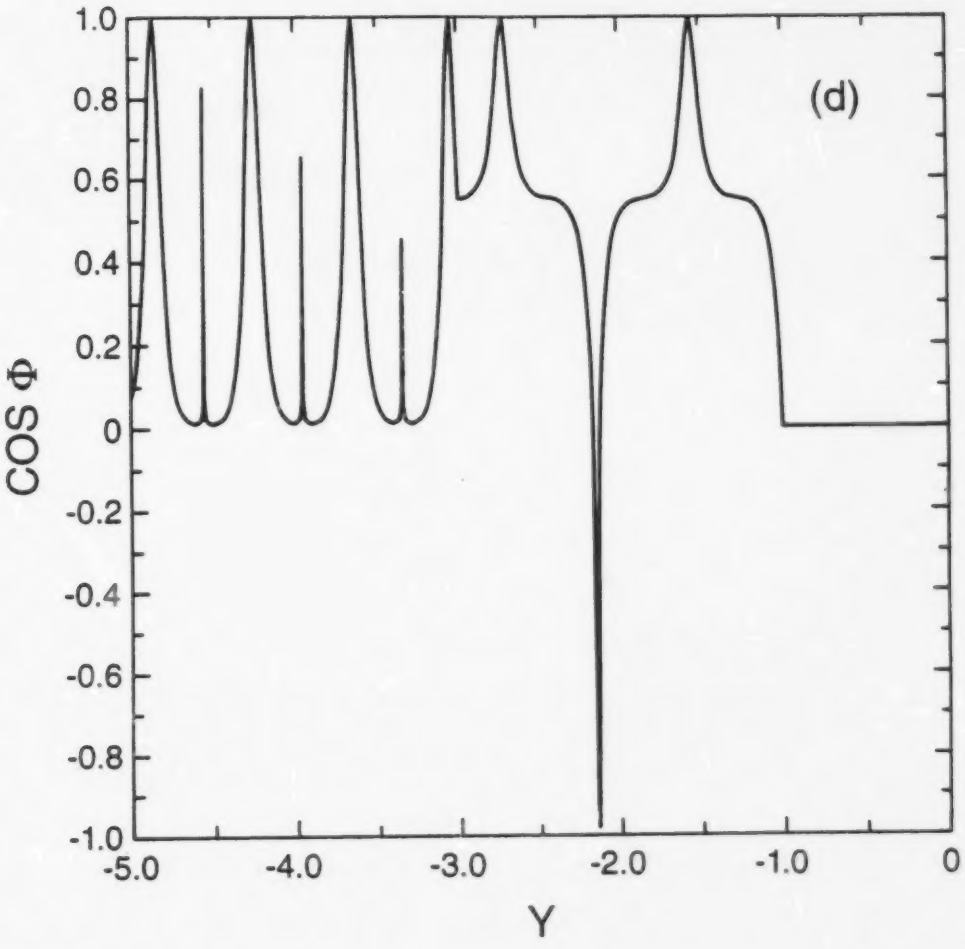
(C) $\Phi_j - \mu_{j0} h_1(Y)/c_{gj}$ versus Y

(D) $\cos(\Phi)$ versus Y

Wave 1, wave 2 and wave 3 are denoted by a thick curve, dashed curve and a dotted curve respectively.


(b)
60
50
40
30
20
10
0
-10
-20
-30
ϕ
-5.0
-4.0
-3.0
-2.0
-1.0
0
Y


(c)
8
7
6
5
4
3
2
1
0
-1
-2
Φ
-5.0
-4.0
-3.0
-2.0
-1.0
0
Y


(d)
1.0
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1.0
COS Φ
-5.0
-4.0
-3.0
-2.0
-1.0
0
Y

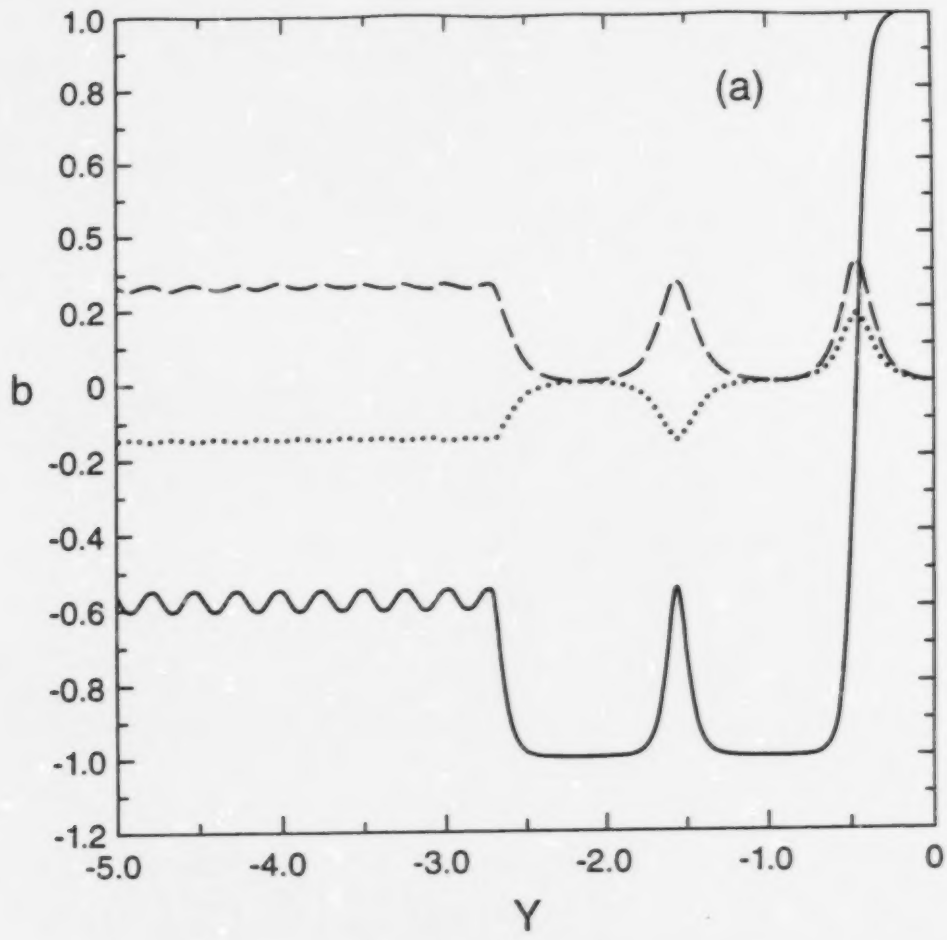


Fig. 8. Plots for initial amplitudes (6.2.10), initial phases (6.2.9), and topography given by (6.2.1) with $H = 0.5$, $Y_0 = -1.0$, and $Y_1 = -2.72$.

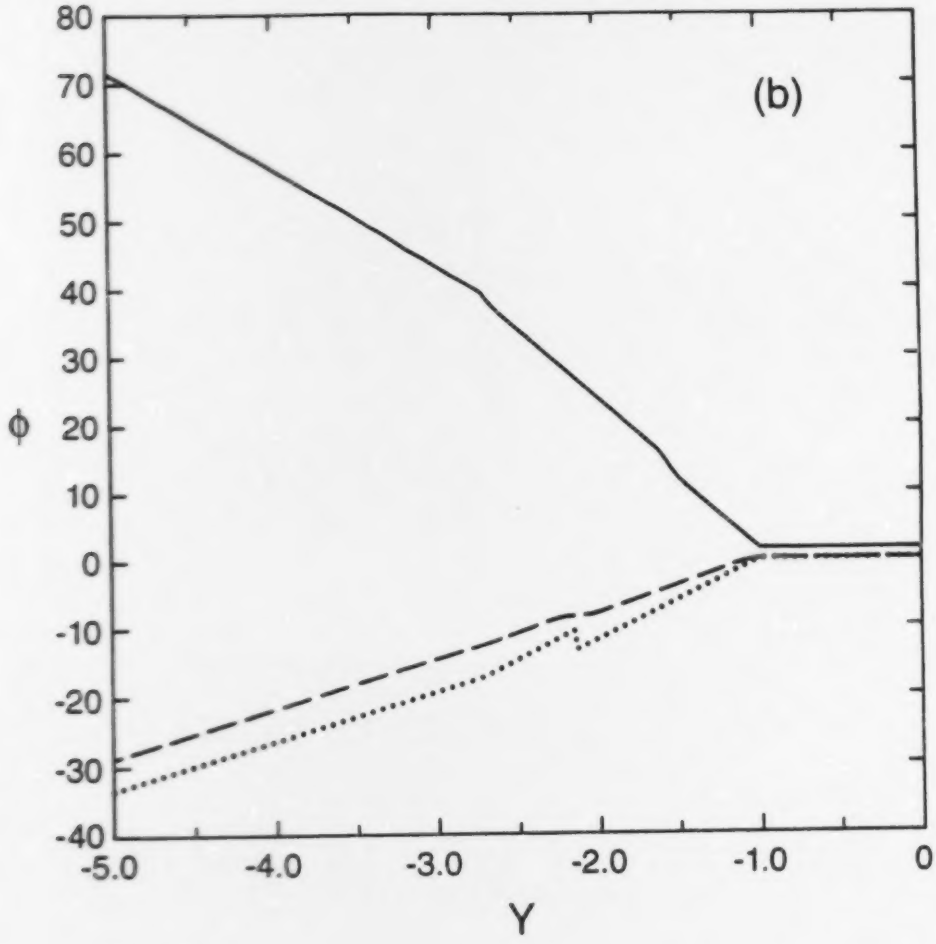
(A) b_j versus Y

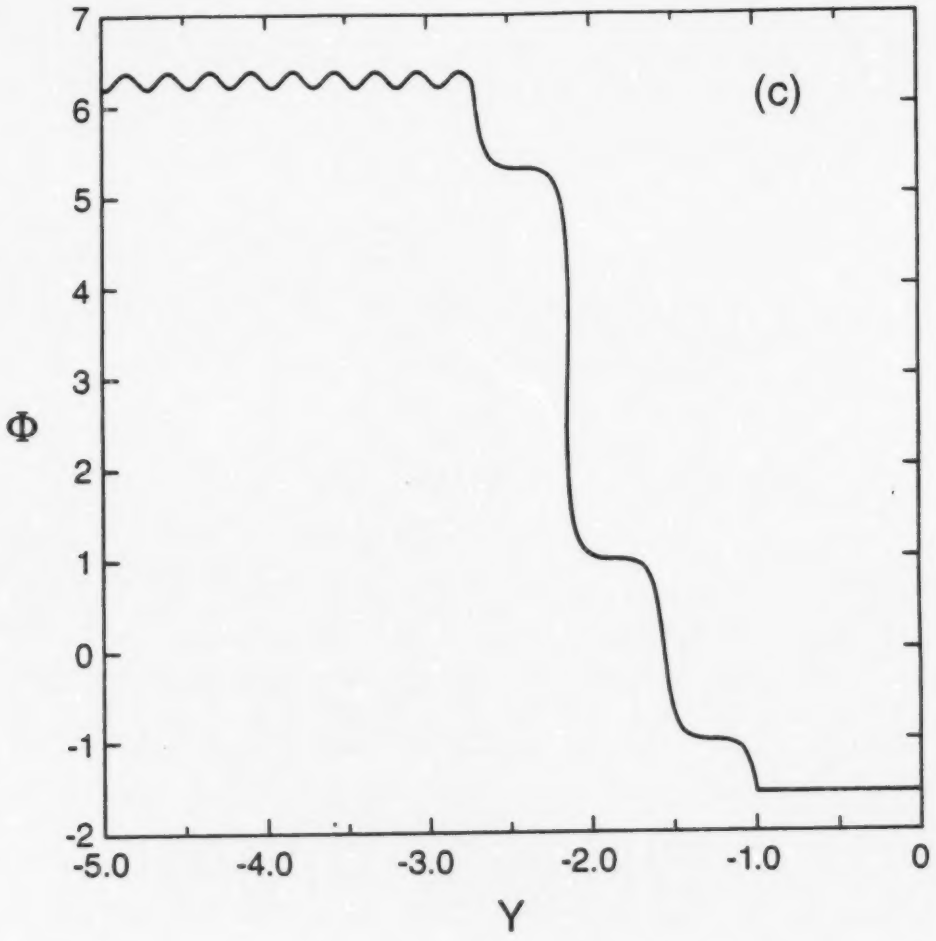
(B) Φ versus Y

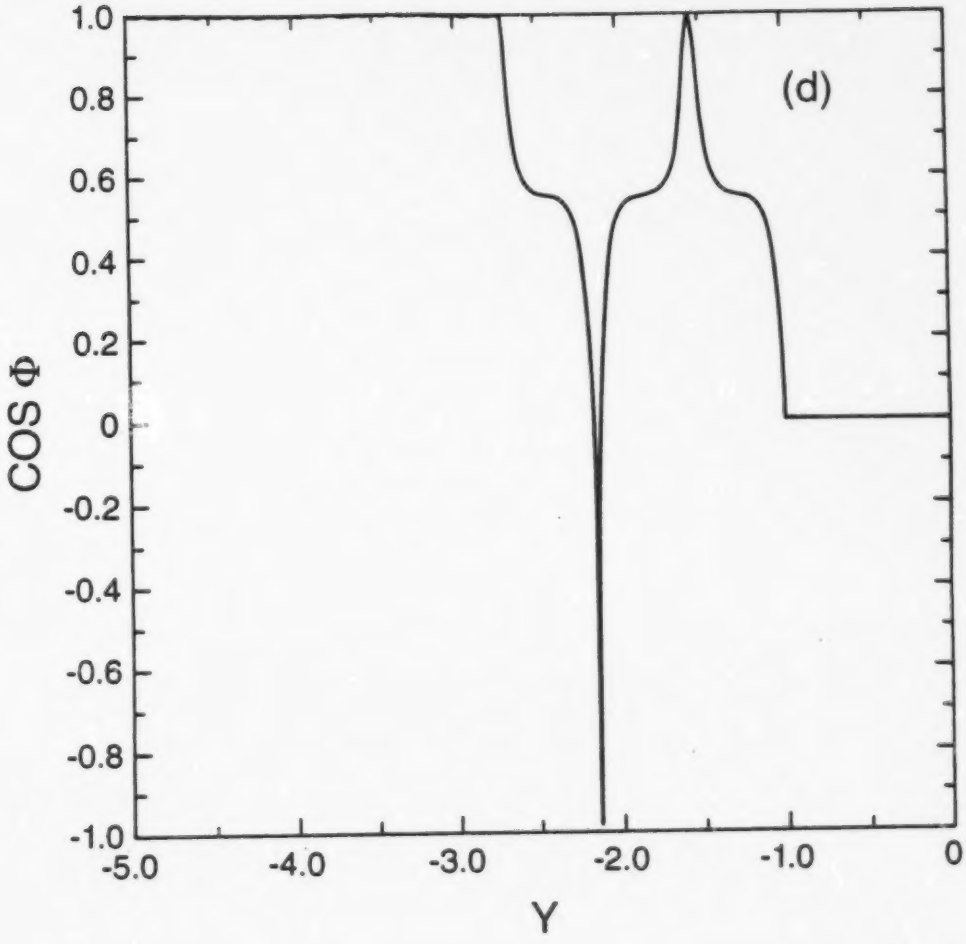
(C) $\Phi_j - \mu_{j0} h_1(Y)/c_{gj}$ versus Y

(D) $\cos(\Phi)$ versus Y

Wave 1, wave 2 and wave 3 are denoted by a thick curve, dashed curve and a dotted curve respectively.


(b)
80
70
60
50
40
30
20
10
0
-10
-20
-30
-40
ϕ
-5.0
-4.0
-3.0
-2.0
-1.0
0
Y


(c)
7
6
5
4
3
2
1
0
-1
-2
Φ
-5.0
-4.0
-3.0
-2.0
-1.0
0
Y


(d)
1.0
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1.0
COS Φ
-5.0
-4.0
-3.0
-2.0
-1.0
0
Y

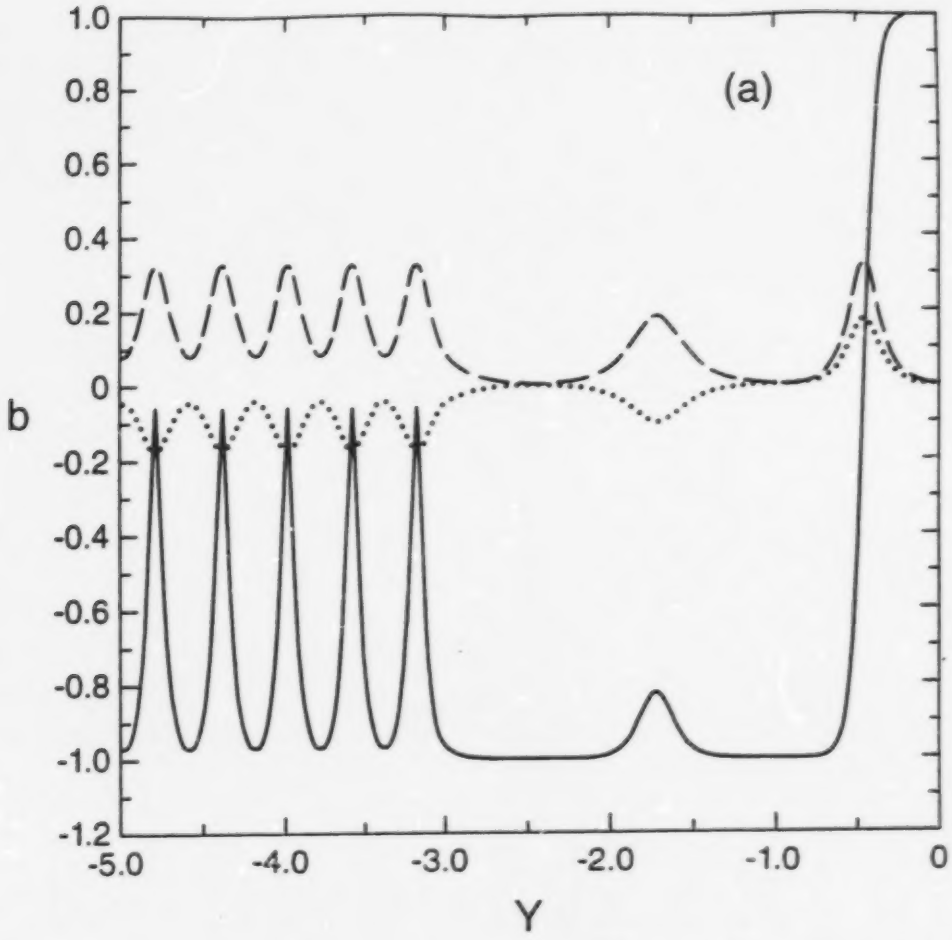


Fig. 9. Plots for initial amplitudes (6.2.10), initial phases (6.2.9) and topography given by (6.2.1) with $H = 0.75$, $Y_0 = -1.0$ andc $Y_1 = 3.0$

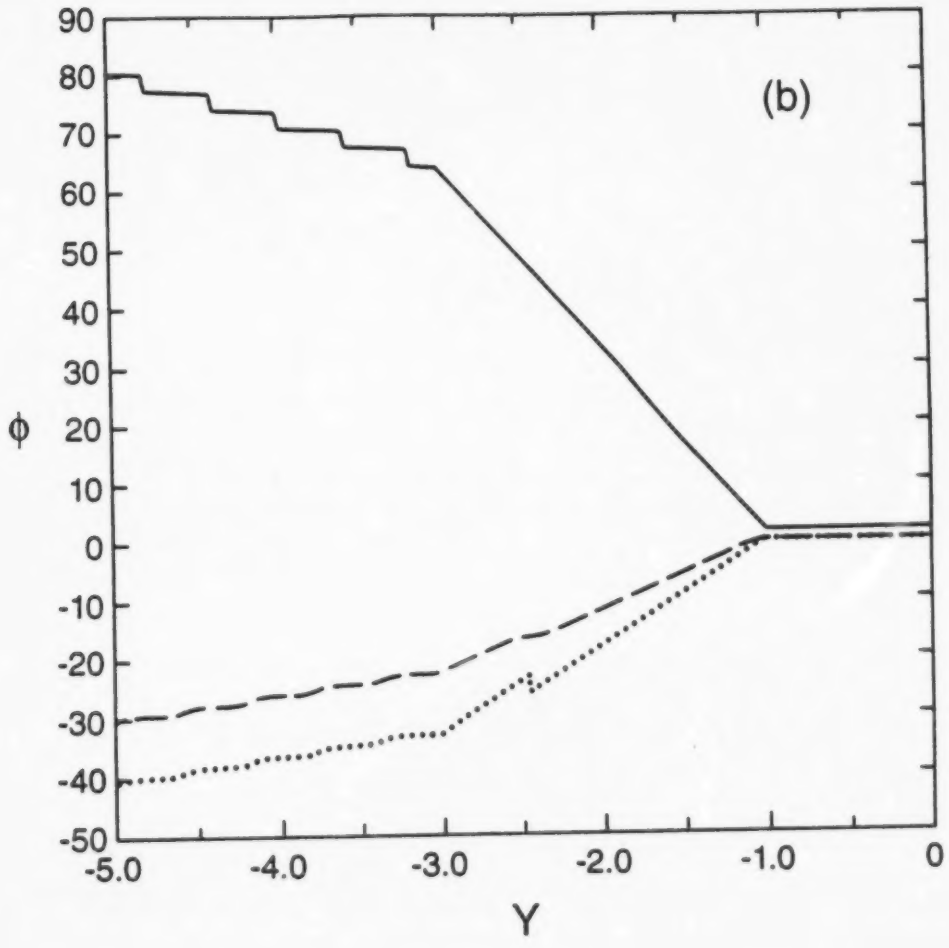
(A) b_j versus Y

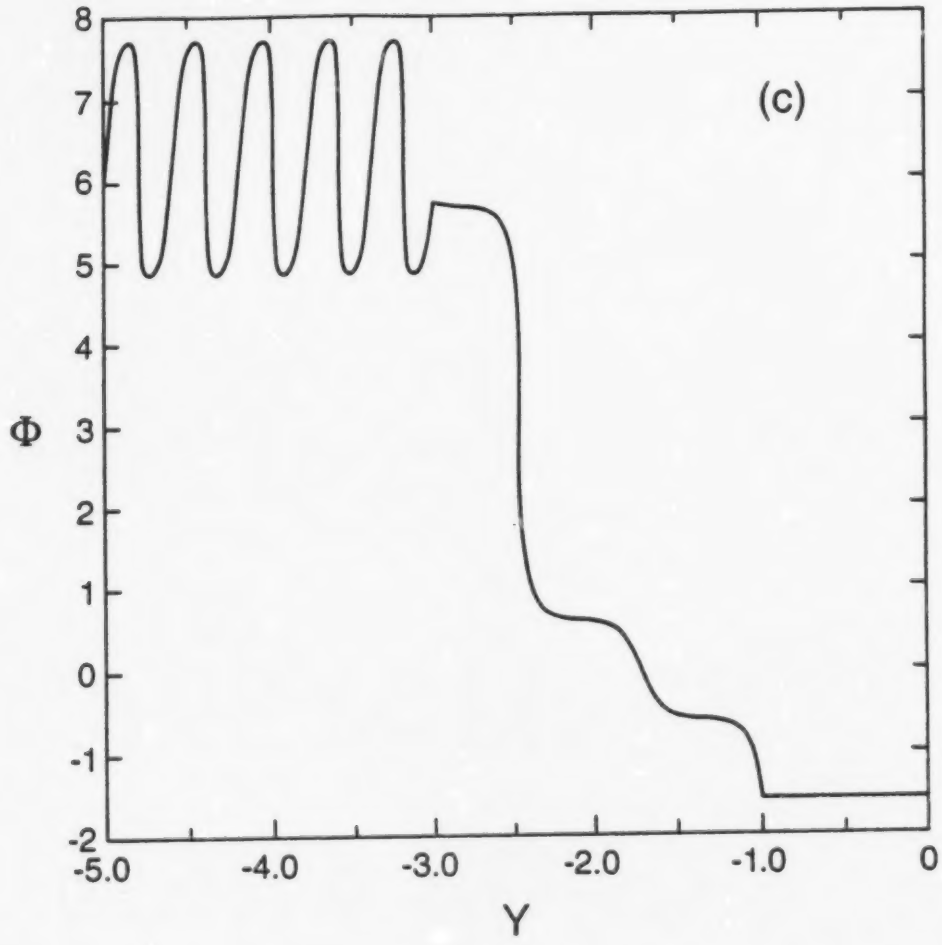
(B) Φ versus Y

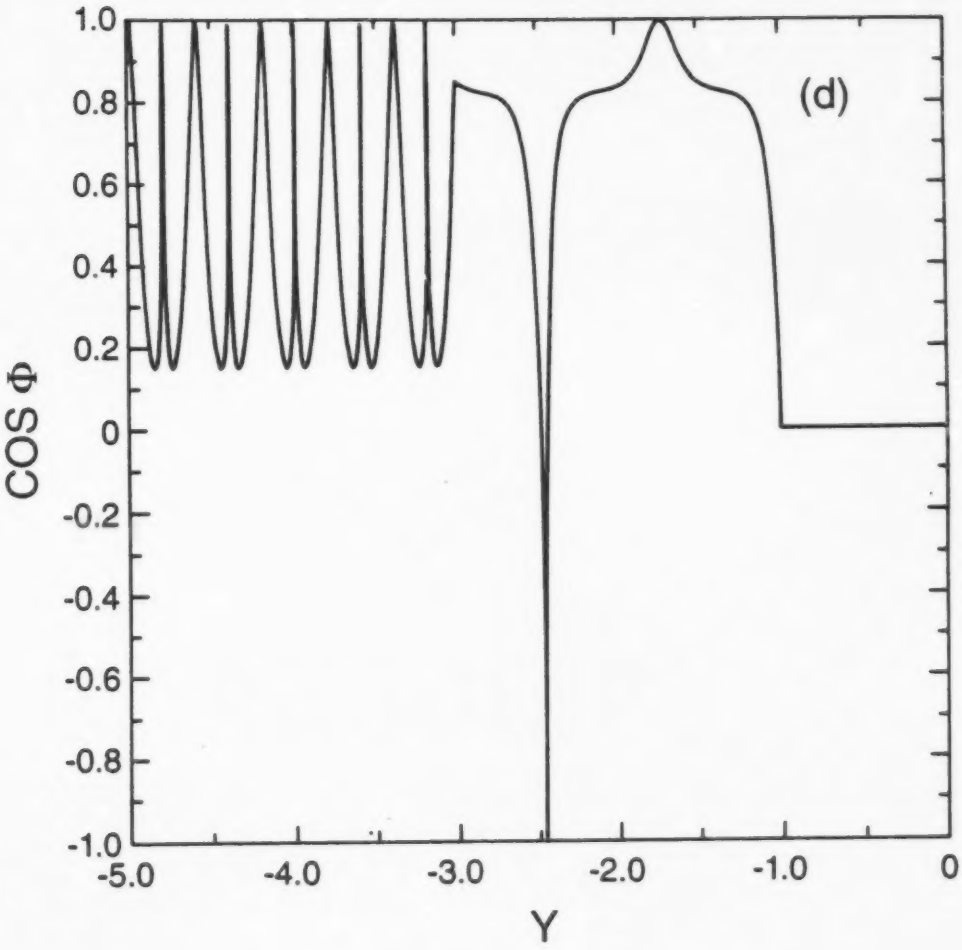
(C) $\Phi_j - \mu_{j0} h_1(Y)/c_{gj}$ versus Y

(D) $\cos(\Phi)$ versus Y

Wave 1, wave 2 and wave 3 are denoted by a thick curve, dashed curve and a dotted curve respectively.


(b)
90
80
70
60
50
40
30
20
10
0
-10
-20
-30
-40
-50
ϕ
-5.0
-4.0
-3.0
-2.0
-1.0
0
Y


(c)
8
7
6
5
4
3
2
1
0
-1
-2
Φ
-5.0
-4.0
-3.0
-2.0
-1.0
0
Y


(d)
1.0
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1.0
COS Φ
-5.0
-4.0
-3.0
-2.0
-1.0
0
Y

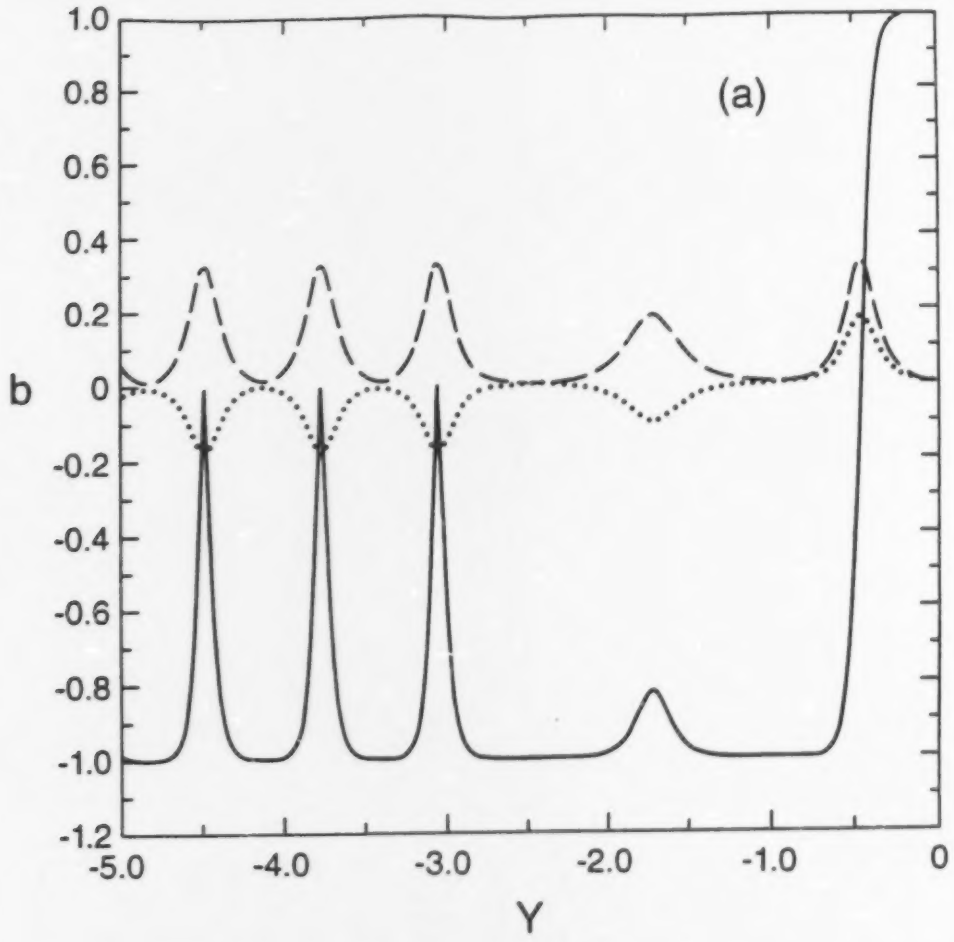


Fig. 10. Plots for initial amplitudes (6.2.10), initial phases (6.1.9) and topography given by (6.2.1) with $H = 0.75$, $Y_0 = -1.0$ and $YH_1 = -2.72$.

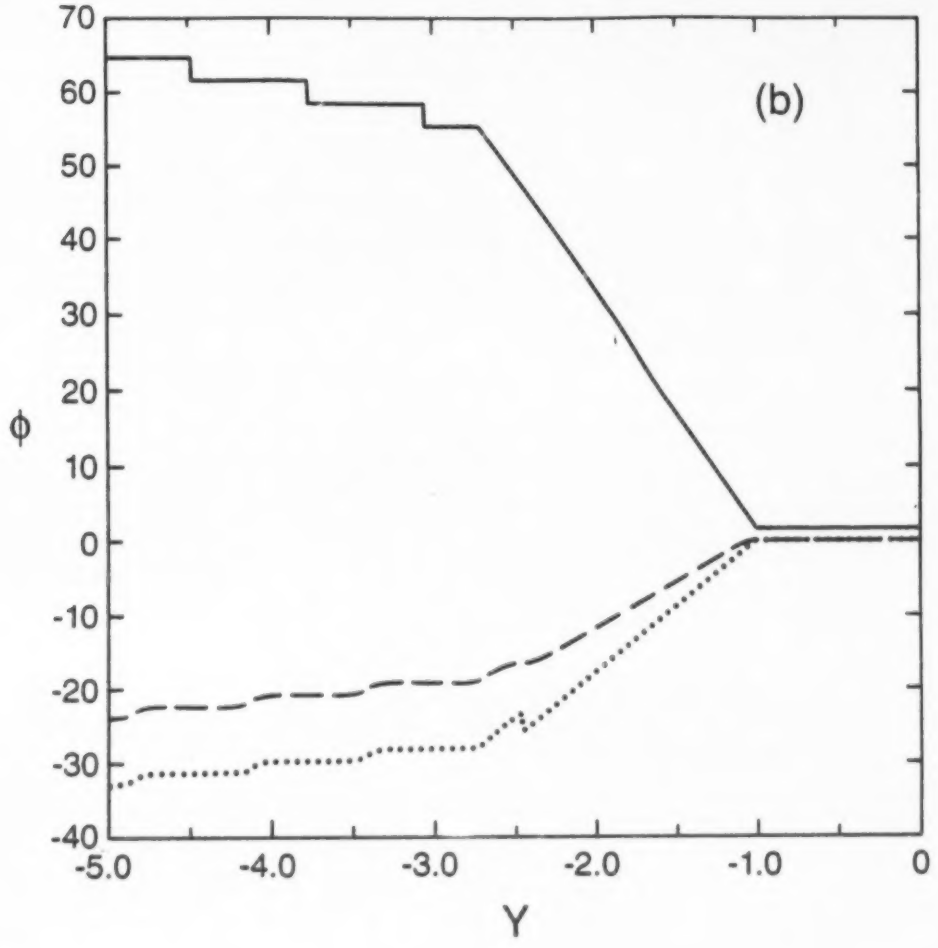
(A) b_j versus Y

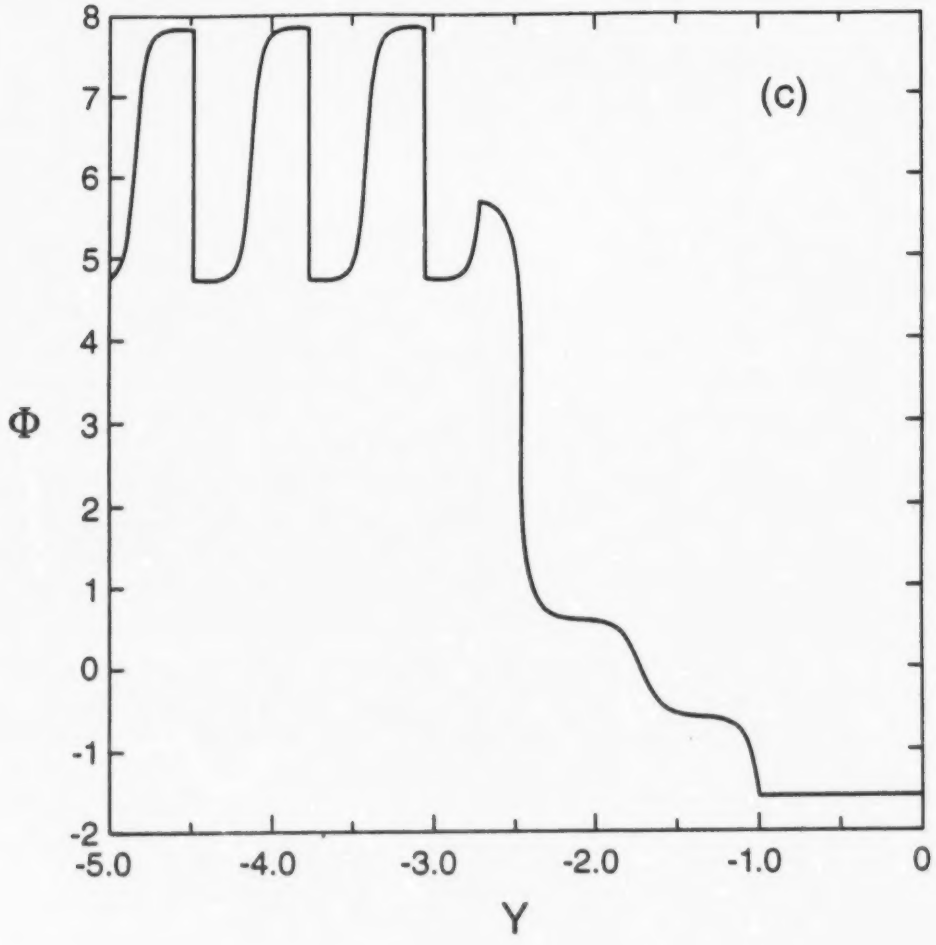
(B) Φ versus Y

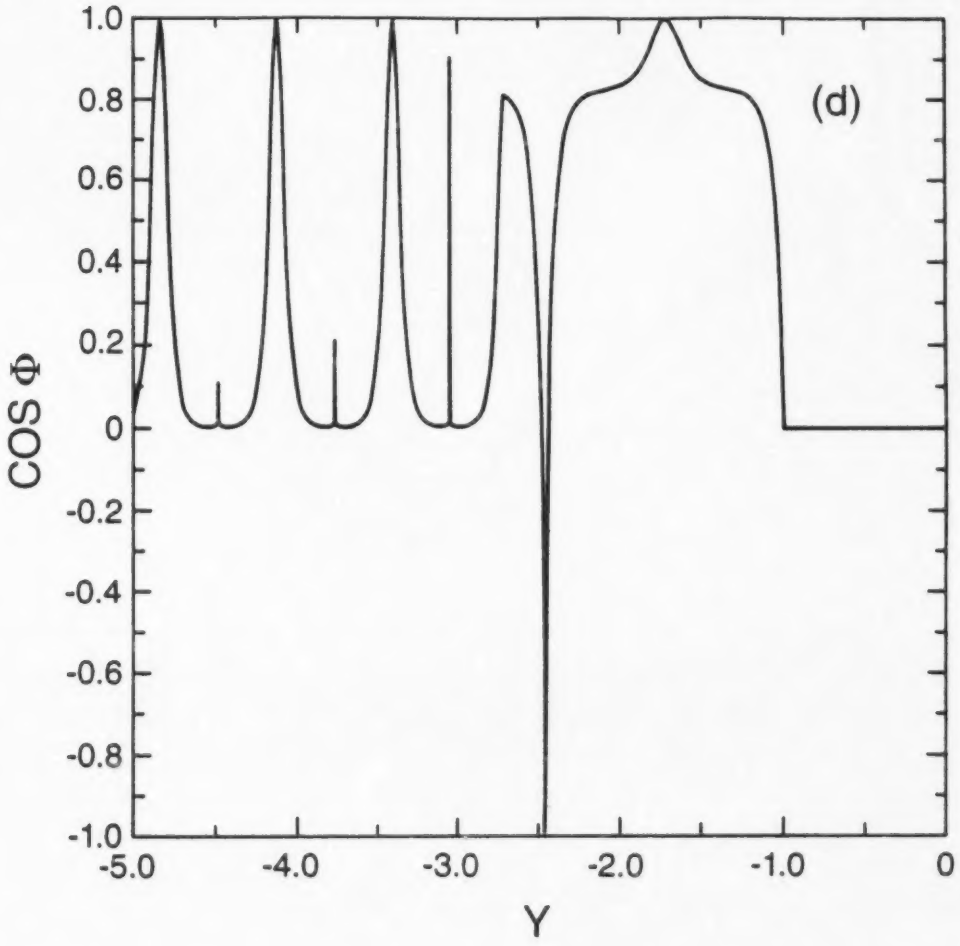
(C) $\Phi_j - \mu_{j0} h_1(Y)/c_{gj}$ versus Y

(D) $\cos(\Phi)$ versus Y

Wave 1, wave 2 and wave 3 are denoted by a thick curve, dashed curve and a dotted curve respectively.


(b)
70
60
50
40
30
20
10
0
-10
-20
-30
-40
ϕ
-5.0
-4.0
-3.0
-2.0
-1.0
0
Y


(c)
8
7
6
5
4
3
2
1
0
-1
-2
Φ
-5.0
-4.0
-3.0
-2.0
-1.0
0
Y


(d)
COS Φ
Y
1.0
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1.0
-5.0
-4.0
-3.0
-2.0
-1.0
0

Chapter 7

Conclusion

We have derived an analytical asymptotic theory which describes the propagation of a triad of continental shelf waves under the influence of Ekman friction and slowly varying alongshore topography. We have assumed that the water depth change due to alongshore topography, is of the same order of magnitude as the nonlinear terms in the vorticity equation. Since no exact solutions can be found for the purely temporal problem with nonzero Ekman friction, we illustrated numerical solutions in the absence of alongshore topographic variation. For the steady-state problem in the presence of piecewise constant topographic configurations, and in the absence of friction, we have shown that the perturbed interaction equations may be solved exactly.

We began by formulating the problem in terms of the nonlinear barotropic long-wave equations with Ekman friction terms, for a rotating system. In order to highlight the essential features of the triad interaction we neglected surface windstress and made the nondivergence approximation. To study the effects of bottom friction and alongshore topographic variations we looked at the purely-temporal problem and the steady-state problem separately.

For the purely temporal problem with bottom friction and no alongshore topography our conclusions are as follows

1) The damping coefficent of each member of the triad is proportional to the reciprocal of its phase velocity with the constant of proportionality given by half the product of the coriolis parameter with the Ekman number.
2) The effect of damping due to bottom friction is to modulate the amplitude of each wave with an exponentially decaying envelope.
3) Because of the large variablity in the estimates for the Ekman number it is impossible to determine whether bottom friction will completely inhibit resonant interactions. For the large estimates of E_V damping eliminates resonant

interactions, but for the lower estimates fo E_V resonant energy exchange is still possible.

For the steady state solution in the absence of bottom friction our conclusion are as follows:

4) In the absence of topographic forcing, the magnitude of the wave with the interaction coefficient which has a sign opposite to the other two will always vary in the opposite direction to the other two, and in this way the wave energy will oscillate between the modes. This oscillation shows reversible behavior.

5) In the presence of topography, the waves behave as in 4, except the strength of the interaction is less, due to a dephasing of the wave as they propagate over the topography. This dephasing is still present after the waves have propagated over the topography.

6) The dephasing is caused by an alongshore wave number mismatch which is induced by the topography. The size of the wavenumber mismatch and hence the magnitude of the dephasing is dependent on ε, the height and position of the topography and the cross shelf profile.

7) Different configurations of piecewise linear topography have analogous effects on the triad.

The theory presented in this thesis may be improved in several ways. We have presented a theory for discontinuous piecewise constant topographic feature, which served to illustrate in a simple way the qualitative effect of alongshore topography variation. It would be interesting to numerically solve the problem for a more realistic alongshore topographic configuration modeling the Oregon coast, so that the results may be compared to the observational data obtained by Cutchin and Smith (1973).

Also the resonant triad used in Hsieh and Mysak (1980) could not be studied via our steady state model since that triad depended on both space and time

(the group velocities differed in sign). Perhaps asymptotic solutions that include variations in both space and time could be obtained by the inverse scattering method.

Since we have seen that nearly resonant waves can exchange energy, it would be interesting to study how a spectrum of initially nearly resonant triads evolves over a topographic feature, to see how the resonant interactions will distribute the energy over the entire spectrum.

Bibliography

Abramowitz, M. and J.A. Stegun, 1965: Handbook of Mathematical Functions. Dover, New York.

Adams, J.K. and V.T. Buchwald, 1969: The generation of continental shelf waves. J. Fluid Mech., 35, 815-826.

Allen, J.S. 1980: Models of wind-driven currents on the continental shelf. Ann. Rev. Fluid Mech. 12, 389-433.

Bender, C.M. and Orzag, S.A., 19878: Advanced Mathematical Methods for Scientists and Engineers. McGraw,Hill.

Boyce, W.E. and R.C. DiPrima, 1969: Elementary Differential Equations and Boundary Value Problems, 2nd ed. Wiley, 533 pp.

Brink, K.H., 1991: Coastal-trapped waves and wind-driven currents over the continental shelf. Ann. Rev. Fluid Mech. 23, 389-412.

Brink, K.H. and Allent, J.S. 1978: On the effect of bottom friction on barotropic motion over the continental shelf. J. Phys. Oceanogr. 8, 919-922.

Buchwald, V.T. and J.K. Adams, 1968: The propagation of continental shelf waves. Proc. Roy. Soc. London, A305, 235-250.

Chang, P., 1991: Coastal Kelvin waves in the presence of a slowly varying topography. J. Fluid Mech. Vol 000, pp. 000-000.

Craik, A.D.D., 1985: Wave interactions and fluid flows. Cambridge university Press.

Cree, W. 1990: Topographic de-phasing and amplitude modulation of Rossby wave triad packets. M.Sc. Thesis, Univerity of Alberta, Edmonton, Canada.

Cree, W. and G.E. Swaters, 1991: On the topographic dephasing and amplitude modulation of nonlinear Rossby wave interactions. Geophys. Astrophys. Fluid Dynamics., 61 , 67-99.

Cutchin, D.L. and R.L. Smith, '1973: Continental shelf waves: Low-frequency variations in sea level and currents over the Oregon continental shelf. J. Phys. Oceanogr. 3, 73-82.

Dorr, A. and R. Grimshaw, 1990: Continental shelf wave scattering by a rigid barrier normal to the coast. J. Phys. Oceanogr. 20, 1849-1866.

Grimshaw, R., 1977a: Nonlinear aspects of long shelf waves. Geophys. Astrophys. Fluid Dyn., 8, 3-16.

Grimshaw, R., 1977b: The stability of continental shelf waves. I. Side band instability and long wave resonance. J. Aust. Math. Soc., B20, 13-30.

Grimshaw, R., 1977c: The Effects of a Variable Coriolis Parameter, Coastline Curvature and Variable Bottom Topography on Continental Shelf Waves. J. Phys. Oceanogr. 7, 547-554.

Hsieh, W.W. and L.A. Mysak, 1980: Resonant interactions between shelf waves, with applications to the Oregon Shelf. J. Phys. Oceanogr., 10(11), 1729-1741.

Hayer, A.B., M. Hickey, J.D. Smith, R.L. Smith and R.D. Pillsbury, 1975: Alongshore coherence at low frequency in currents observed over the continental shelf off Oregon and Washington. J. Geophys. Res., 80, 3495-3505.

Kundu, P.K. and J.S. Allen, 1976: Some three-dimensional characteristics of low-frequency current fluctuations near the Oregon coast. J. Phys. Oceanogr., 6, 181-199.

Leblond, P.H. and L.A. Mysak, 1978: Waves in the Ocean. Elsevier, 602pp.

Long, B. and P. Chang, 1990: Propogation of an equitorial Kelvin wave in a varying thermocline. J. Phys. Oceanogr., 20, 1826-1841.

Longuet-Higgins, M.S. 1972: Topographic Rossby waves. Mem. Soc. R. Sci. Liege. 6e Ser 2:11-16.

Mysak, L.A., 1980a: Recent advances in shelf wave dynamics. Rev. Geophys. Space. Phys., 18, 211-241.

Mysak, L., 1980b: Topographically trapped waves. Ann. Rev. Fluid Mech. 12, 45-76.

Pedlosky, J. 1987: Geophysical Fluid Dynamics. Springer-Verlag.

Smith, R., 1972: Nonlinear Kelvin and continental shelf waves. J. Fluid Mech., 52, 379-391.

Swaters, G.E., 1986: Barotropic modon propagation over slowly varying topography. Geophys. Astrophys. Fluid Dynamics, 36, 85-113.

Weiland, J. and H. Wilhelmsson, 1977: Coherent nonlinear interactions of waves in plasmas. Pergamon.

Appendix 1

The Group Velocity

In this appendix we will show that the c_{gj} defined in (2.2.22) is indeed the group velocity of the jth wave, i.e.

$$\frac{\partial w_j}{\partial k_j} = c_{gj} \,. \tag{A1.1}$$

Proof: The proof we give is the same as the one given in the appendix of the paper by Hsieh and Mysak (1980). We reproduce their proof with complete details for the readers convenience. We start with equation (2.2.2), which we multiply by ϕ_j and replace c_j by ω_j/k_j to obtain

$$-\phi_j(\frac{-1}{h_0}\phi_j') + \frac{-k_j^2}{h_0}\phi_j^2 - \frac{k_j}{\omega_j}(\frac{h_0'}{h_0^2})\phi_j^2 = 0. \tag{A1.2}$$

We can now integrate (A1.2) w.r.t. x from 0 to ∞ to yield

$$\int_0^\infty [\phi_j \frac{\partial}{\partial k_j}(\frac{1}{h_0}\phi_j')' - (\frac{1}{h_0}\phi_j')'\frac{\partial \phi_j}{\partial k_j}]dx - \int_0^\infty 2\frac{k_j}{h_0}\phi_j^2 dx$$
$$- \int_0^\infty \frac{1}{\omega_j}(\frac{h_0'}{h_0^2}\phi_j^2 dx + \int_0^\infty \frac{k_j}{\omega_j^2}\frac{\partial \omega_j}{\partial k_j}(\frac{h_0'}{h_0^2}\phi_j^2 dx = 0 \tag{A1.3}$$

If we denote the first integral on the left hand side of (A1.3) by I, and use integration by parts on I we obtain

$$I = [\phi_j \frac{\partial}{\partial k_j}(\frac{1}{h_0}\phi_j') - \frac{\partial \phi_j}{\partial k_j}(\frac{1}{h_0}\phi_j')]|_0^\infty$$
$$- \int_0^\infty \phi_j' \frac{\partial}{\partial k_j}(\frac{1}{h_0}\phi_j') - \frac{\partial \phi_j}{\partial k_j}(\frac{1}{h_0}\phi_j')dx. \tag{A1.4}$$

Applying the boundary conditions (2.2.3) and (2.2.4), I vanishes to that (A1.3) becomes

$$\frac{\partial \omega_j}{\partial k_j}\int_0^\infty \frac{h_0'}{h_0^2}\phi_j^2 dx = \frac{\omega_j}{k_j}\int_0^\infty \frac{h_0'}{h_0^2}\phi^2 - j dx + 2\omega_j^2 \int_0^\infty \frac{1}{h_0}\phi_j^2 dx. \tag{A1.5}$$

Using the normalization condition (2.2.19), and (2.2.23), equation (A1.5) simplifies to

$$\frac{\partial \omega_j}{\partial k_j} = c_j(1 + 2\gamma_j c_j k_j^2), \tag{A1.6}$$

as required.

```
/*********************************************************/
/* Name: buc                                             */
/*                                                       */
/* Purpose: Calculate and plot the first four mode       */
/*          dispersion relations for the exponential     */
/*          shelf profile as described in the paper      */
/*          by  Buchwald and Adams,                      */
/*          Proc. Roy. Soc. A. 305 235-250 (1968).       */
/*                                                       */
/* Written: Feb 13,1992                                  */
/* By: Francois Primeau                                  */
/*                                                       */
/*********************************************************/
#include <stdio.h>
#include <math.h>
#define BASE_K  0.382 /*base wave number used to find roots*/
#define LOWER 0.0     /*left endpoint of interval over which
                        to plot the   dispersion relation */
#define UPPER 10.0    /*right endpoint of interval over which
                        to plot the dispersion relation */
#define STEP 100      /*number of steps at which to calculate
                        the dispersion relation */
#define MODES 4       /*number of modes to plot */

static double k;         /*wave number */
static double omega;     /*frequency */
static double b = 1.65; /*b characterizes the exp. profile */

main()
{
   FILE  *out;                /*out is the output file */
   extern double f();         /*transcendental function*/
   const double tol = 10e-10;/*absolute error tolerance */
   double right, left;        /*interval to look for roots*/
   double zeroin_();          /*FORTRAN zero finding function*/
   double z;                  /*root of transcendental
                                equation*/
   float Y[MODES][STEP];      /*ordinate for graph*/
   float X[STEP];             /*abscissa for graph*/
   int n = 0;                 /*mode number*/
   int i;                     /*point index*/
   double step_size;

   step_size =  ( UPPER - LOWER )/ STEP;

   out = fopen("data","w");   /*open the data output file*/
```

```
/*for each mode calculate the dispersion relation*/
while ( n < MODES)
{
  fprintf(out,"mode (%d)\n",n+1); /* print the root*/

  k = BASE_K;/* initialize wave number*/
  fprintf(out,"wave number k = %lg \n",k);

  i = 0;
  k = 0;
  while ( i < STEP )
  {
     left = (n + 0.5)*M_PI+1.0e-9;  /*set left endpoint*/
     right = (n + 1.5)*M_PI-1.0e-9; /*set right endpoint*/
     z = zeroin_(&left,&right,f,&tol); /*find the root*/

     omega = -2*b*k/(z*z+k*k+b*b); /*compute the frequency*/
     X[i] = (float)k;
     Y[n][i] = (float)omega;
     fprintf(out," %d  wave number: %lg, frequency: %lg \n",i,
             X[i],Y[n][i]);
     k = k + step_size;
     i++;
  }
  n++;
}

/*make the graph of the dispersion relation*/
{
  static char xnote[]="wave number$";
  static char ynote[]="frequency$";
  static char title[]="Dispersion Relation$";
  int ier;
  const int n_mode = MODES;
  const int size = STEP;
  const int _0 = 0;
  const int _1 = 1;
  const int _2 = 2;
  const int _6 = 6;
  gopks_(&_6,&ier);
  gopwk_(&_1,&_2,&_1);
  gacwk_(&_1);
  anotat_(xnote,ynote,&_1,&_1,&_0,&_0);
  ezmxy_(X,Y,&size,&n_mode,&size,title);
  gdawk_(&_1);
  gclwk_(&_1);
  gclks_();
```

```
    }
}
double f(double *z) /* transcendental function */
{
        return(tan(*z)+*z/(b +_ABS(k)));
}
```

```
/*****************************************************************/
/* Name: m_triad                                                 */
/*                                                               */
/* Purpose: Find a resonant triad given a first mode             */
/*          wave number.                                         */
/*                                                               */
/* Written: Mar 4,1992                                           */
/* By: Francois Primeau                                          */
/*****************************************************************/
#include <stdio.h>
#include <math.h>
#define SHELF_PARAMETER 1.65
#define BASE_WAVE_NUMBER 0.382
main()
{
   double k[4],omega[4],z[4];
   const double b = SHELF_PARAMETER;
   int i;
   extern int triad();

   FILE  *out;                  /*out is the output file */
   out = fopen("triad.dat","w");/*open the data output file*/
   fprintf(out,"\n \n");
   fprintf(out,"                          Resonant Triad      \n");
   fprintf(out,"                          --------------      \n");
   fprintf(out,"      ( wave number, frequency ), root \n \n");

   k[1] = 0.3;
   while (k[1] < 0.5  ) {

      i = triad(k,omega,z,b);

      fprintf(out,"      (%.10lf,%.10lf) z1 = %.10lf \n",k[1],
            omega[1],z[1]);
      fprintf(out,"      (%.10lf,%.10lf) z2 = %.10lf \n",k[2],
            omega[2],z[2]);
      fprintf(out,"      (%.10lf,%.10lf) z3 = %.10lf \n \n",
            k[3],omega[3],z[3]);
   k[1] = k[1] + 0.001;
   }
}
```

```
/*****************************************************************/
/* Name: triad                                                   */
/*                                                               */
/* Purpose: Find a resonant triad given a first mode wave        */
/*          number.                                              */
/*                                                               */
/* Written: Mar 4,1992                                           */
/* By: Francois Primeau                                          */
/*                                                               */
/*****************************************************************/
#include <stdio.h>
#include <math.h>

#define TOL 1.0e-10 /*absolute error tolerence*/

static double global_k;       /* wave number*/
static double b;
int triad(double k[4],
      double omega[4],
      double z[4],
      double shelf_parameter )
{
   double k1,k2,k3;        /*resonant triad of wave numbers*/
   double om1,om2,om3;     /*resonant triad of frequencies*/
   double z1,z2,z3;        /*roots of transcendental equation*/
   extern double disp();   /*dispersion relation*/
   extern double f();      /*transcendental function*/
   double left1,right1;    /*interval to look for first mode
                             root*/
   double left2,right2;    /*interval to look for second mode
                             root*/
   double zeroin_();       /*FORTRAN zero finding function*/
   const double tol = TOL;/*absolute error tolerence*/
   double step = 1.0e-2;
   double test1 = 0.0;
   double test2 = 0.0;
   int cont;

   left1= (0.5)*M_PI+1.0e-9;/*set left endpoint for second
                              mode*/
   right1 = (1.5)*M_PI-1.0e-9;/*set right endpoint for second
                               mode*/

   left2 = (1.5)*M_PI+1.0e-9;/*set left endpoint for first
                              mode*/
   right2 = (2.5)*M_PI-1.0e-9;/*set right endpoint for first
                               mode*/
```

```
b = shelf_parameter;
k1 = k[1];
global_k = k1;
z1 = zeroin (&left1,&right1,f,&tol);/* find the root*/
om1 = disp(k1,z1);

k2 = 0.0;
cont = 1;
while(cont)
{
   k2 = k2 + step;
   global_k = k2;
   z2 = zeroin (&left2,&right2,f,&tol);/* find the root*/
   om2 = disp(k2,z2);

   k3 = k1 + k2;
   global_k = k3;
   z3 = zeroin (&left1,&right1,f,&tol);/* find the root*/
   om3 = disp(k3,z3);

   test1 = om3-(om1+om2);
   if ( test2 == 0 ) test2 = test1;
   else if (test1*test2/_ABS(test1*test2) > 0.0)
        {
           test2 = test1;
        }
        else if (_ABS(test1) >= tol)
             {
               k2 =k2 - step;
               step = step/10;
             }
             else
             {
                cont = 0;
             }
}
k[2] = k2;
k[3] = -k3;
omega[1] = om1;
omega[2] = om2;
omega[3] = -om3;
z[1] = z1;
z[2] = z2;
z[3] = z3;

printf("k: %le %le %le \n",k[1],k[2],k[3]);
printf("w: %le %le %le \n",omega[1],omega[2],omega[3]);
```

```
   printf("z: %le %le %le \n",z[1],z[2],z[3]);
   return(0);
}
double f(double *z) /* transcendental function */
{
  return(tan(*z)+*z/( b +_ABS(global_k)));
}
double disp(double k,double z)
{
  return(-2*b*k/(z*z+k*k+b*b));
}
```

Appendix 4

```
/*****************************************************************/
/* Name: friction                                                */
/*                                                               */
/* Purpose: This program numerically intergrates the three       */
/*          wave interaction equations with friction.            */
/*                                                               */
/* Written: Mar 5,1992                                           */
/* By: Francois Primeau                                          */
/*                                                               */
/*****************************************************************/
#include <stdio.h>
#include <math.h>
#define SHELF_PARAMETER 1.65  /*exponential shelf parameter*/
#define H2_ 14.2              /*nondimentionalized deep sea
                                depth*/
#define BASE_WAVE_NUMBER 0.382/*first mode wave number*/
#define R_ 0.79               /*friction parameter*/
#define AMP10 1               /*first wave initial amplitude*/
#define AMP20 0.6             /*second wave initial amplitude*/
#define AMP30 0.0             /*third wave initial amplitude*/
#define AERR 1.0e-8           /*absolute error tolerance*/
#define RERR 1.0e-8           /*relative error tolerance*/
#define STEP_SIZE 0.0006
#define EPSILON 0.008928571
#define CORIOLIS_PARAMETER 1.0E-4
FILE *out;
static double KK[4];          /*interaction coefficients*/
static double r[4];           /*friction coefficients*/
static double K[4][4][4] = {
          {{0,0,0,0},{0,0,0,0},{0,0,0,0},{0,0,0,0}},
          {{0,0,0,0},{0,0,0,0},{0,0,0,0},{0,0,0,0}},
          {{0,0,0,0},{0,0,0,0},{0,0,0,0},{0,0,0,0}},
          {{0,0,0,0},{0,0,0,0},{0,0,0,0},{0,0,0,0}}};

main()
{
int i;
double k[4];/*wave numbers*/
double omega[4];/*frequencies*/
double z[4];/*roots of transcendental equation*/
double N[4];/*normalization constants*/
double c[4];/*phase velocities*/
const double H2 = H2_;
const double b = SHELF_PARAMETER;
extern int derivative();

float aerr=AERR;          /*absolute error tolerance*/
```

```
float rerr=RERR;          /*relative error tolerance*/
float T = 0;              /*slow time*/
float Tout = 0;           /*time end point for numerical
                            integration*/
float A[4];               /*amplitudes -- B[0] is not used*/
float AA[3][1000];        /*array of amplitudes to be plotted*/
float TT[1000];           /*array of times to be plotted*/
float wk[184];            /*float work array used by ODE_*/
const int n = 4;          /*dimension of A array*/
int ind = 1;              /*index used by ODE_*/
int iwk[9];               /*int work array used by ODE_*/
FILE *out;                /*output file */

out = fopen("friction.dat","w");/*open the data output file*/

k[1] = BASE_WAVE_NUMBER;
i = triad(k,omega,z,b);  /*find the resonant triad*/
i = Nj(N,b,H2,z);     /*compute the normalization constants*/
i = cj(c,k,omega);        /*compute the phase velocities*/


r[1] = -R_/c[1]; /*compute the first friction coefficient*/
r[2] = -R_/c[2]; /*compute the second friction coefficient*/
r[3] = -R_/c[3]; /*compute the third friction coefficient*/

i = Kjlm(K,b,H2,N,k,z);/*compute the interaction coefficients*/
KK[1] = c[1]*(K[1][2][3] + K[1][3][2]);
KK[2] = c[2]*(K[2][3][1] + K[2][1][3]);
KK[3] = c[3]*(K[3][1][2] + K[3][2][1]);

for (i =0; i <1000; i++) /*start numerical integration of ode's*/
{
   A[1] = AMP10;          /*initialize the amplitudes*/
   A[2] = AMP20;          /*initialize the amplitudes*/
   A[3] = AMP30;          /*initialize the amplitudes*/

   Tout = Tout + STEP_SIZE;/*set the right end point of
                             integration*/

   /*ODE is a FORTRAN differential equation solver */
   /*see "NSWC LIBRARY OF MATHEMATICS SUBROUTINES (1980)"
     pp. 461-463*/
   ode_(derivative,&n,A,&T,&Tout,&rerr,&aerr,&ind,wk,iwk);

   switch (ind) {
   case 1: break;
   case 2: break;
```

```
case 3:
   printf("WARNING: error tolerance too small. T = %g\n",T);
   Tout = T;
   break;
case 4:
   printf("WARNING: MAXNUM steps were performed. T = %g\n",T);
   Tout = T;
   break;
case 5:
   printf("WARNING: Equation appeas to be stiff. T = %g\n",T);
   break;
case 6:
   printf("WARNING: AERR is set to zero. T= %g\n",T);
   Tout = T;
   break;
case 7:
   printf("ERROR: no computations were performed.T = %g\n",T);
   break;
case -1: break;
case -2: break;
case -3:
   printf("WARNING: error tolerance too small. T = %g\n",T);
   Tout = T;
   break;
case -4:
   printf("WARNING: MAXNUM steps were performed. T = %g\n",T);
   Tout = T;
   break;
case -5:
   printf("WARNING: Equation appeas to be stiff. T = %g\n",T);
   Tout = T;
   break;
case -6:
   printf("WARNING: AERR is set to zero. T = %g\n",T);
   Tout = T;
   break;
case -7:
   printf("ERROR: no computations were performed. T = %g\n",T);
   break;
}
AA[0][i] = A[1];
AA[1][i] = A[2];
AA[2][i] = A[3];
TT[i] = Tout/(EPSILON*CORIOLIS_PARAMETER*60*60*24);

fprintf(out,"%.14e %.14e %.14e %.14e \n",TT[i],A[1],
        A[2],A[3]);
```

```
}
   /*make the graph of the dispersion relation*/
   {
     static char xnote[]="time (days)$";
     static char ynote[]="amplitude (nondimensionalized)$";
     static char title[40];
     int ier;
     const int n = 3;
     const int size = 1000;
     const int _0 = 0;
     const int _1 = 1;
     const int _2 = 2;
     const int _6 = 6;
     const double R = R_;
     sprintf(title,"Amplitude -vs- Time: r = %g $",R);

     gopks_(&_6,&ier);
     gopwk_(&_1,&_2,&_1);
     gacwk_(&_1);
     anotat_(xnote,ynote,&_1,&_1,&_0,&_0);
     ezmxy_(TT,AA,&size,&n,&size,title);
     gdawk_(&_1);
     gclwk_(&_1);
     gclks_();
   }
   fclose(out);
}

int derivative(float time,  /*time*/
               float A[4],  /*amplitude*/
               float dA[4]) /*derivative of amplitude
                              w.r.t. time*/
{
dA[0] = 0;
dA[1] = (float) -KK[1]*A[2]*A[3]-r[1]*A[1];
dA[2] = (float) -KK[2]*A[3]*A[1]-r[2]*A[2];
dA[3] = (float) -KK[3]*A[1]*A[2]-r[3]*A[3];
}
```

Appendix 5

```
/*****************************************************************/
/* Name: topo                                                    */
/*                                                               */
/* Purpose: This program plots the solution for a resonant       */
/*          triad of waves  over a piece wise constant           */
/*          topographic configuration.                           */
/*                                                               */
/* Written: Feb 13,1992                                          */
/* By: Francois Primeau                                          */
/*                                                               */
/*****************************************************************/
#include <stdio.h>
#include <math.h>

FILE *out;

main()
{
double h;          /* jump parameter */
double H;          /* alongshore topographic paramter */
double mu0;        /* alongshore topographic coefficients */
double mu[4];      /* alongshore topographic coefficients */
double cg[4];      /* group velocities */
double bb,H1,H2;   /* cross-shelf profile parameters */
double k[4];       /* triad of wave numbers */
double omega[4];   /* triad of frequencies */
double z[4];      /*triad of roots to transcendental equation*/
double b0[4];      /*initial amplitude envelopes */
double b0hat[4];   /*scaled initial amplitude envelopes */
double phi0[4];    /*initial phases*/
double K0[4];      /*interaction coefficients*/
double PHI;        /* phase envelope variable */
double PHI0;       /* initial phae envelope */
double Gamma;      /*constant of the motion*/
double yhat[4];    /*roots of the cubic equation*/
double bhat[4];    /* scaled amplitude envelopes */
double b[4];       /* amplitude envelopes */
double phi[4];     /* phase envelopes */
double w[4];       /* weight of Pi/2 attributed to each inital
                      phase */
double zeta;       /* zeta = Y - Ys */
double zetanot;    /* yhat(zetanot) = 0 */
double Y;          /* alongshore slow variable */
double Yp;         /* right endpoint of current alongshore
                      topographic segment */
double Yn;         /* left endpoint of current alongshore
                      topographic segment */
```

```
double Ys;        /* shift parameter */
double khat;      /* modulus */
double thetahat; /* sn phase shift */
double step;      /* number of Y interval steps at which to
                     calculate the solution */
double step_size;/* size of Y interval steps */
int i,j,n,ierr;
double TEST;
double SN,CN,DN;
double s[4];
double S;
FILE *in;         /* file from which to readin the parameters
                     and coefficients */

   in = fopen("topo.dat","r"); /* open the input file */
   out = fopen("out1.dat","w");/* open the output data file*/

   Yp = 0;  /* start simulation at Y = 0.0 */
   i = init(in,b0,w,K0,&mu0,mu,cg,&bb,&H1,&H2,k,omega,z);
   i = scale(b0hat,b0,K0);
   PHI0 = -1*M_PI/2;
   TEST = -1*M_PI/2;
   phi0[1] = w[1]*PHI0;
   phi0[2] = w[2]*PHI0;
   phi0[3] = w[3]*PHI0;
   s[1] = 1.0;
   s[2] = 1.0;
   s[3] = 1.0;
   S = 1.0;
   while (astopo(in,&Yn,&Ys,&zetanot,&H,&step) == 0 )
   {
     h = H;
     Y = Yp;
     zeta = Yp - Ys;
     printf("H: %.14e Yn: %.14e  Yp: %.14e \n", h,Yn,Yp);
     printf(
       "b0hat[1]: %.14e  b0hat[2]: %.14e  b0hat[3] %.14e \n",
        b0hat[1],b0hat[2],b0hat[3]);
     printf("PHI0: %.14e   cos(PHI0): %.14e \n",PHI0,cos(PHI0));
     Gamma = b0hat[1]*b0hat[2]*b0hat[3]*cos(PHI0)+
             mu0*h*b0hat[1]*b0hat[1]/2;
     printf("Gamma: %.14e \n",Gamma);
     i = cubic(yhat,b0hat[1],b0hat[2],b0hat[3],Gamma,mu0,h);
     printf("y1: %.14e  y2: %.14e  y3: %.14e \n",yhat[1],
             yhat[2],yhat[3]);
     khat = sqrt((yhat[1]-yhat[2])/(yhat[1]-yhat[3]));
     if ( invsn(&thetahat,
```

```
                sqrt(yhat[1]/(yhat[1]-yhat[2])),khat
                )!=0) error(1);
        step_size = (Yn - Yp)/step;
        i = solve(bhat,phi,s,S,&PHI,&TEST,yhat,b0hat,phi0,PHI0,
                  Gamma,mu0,h,khat,thetahat,zeta+ 10*step_size,
                  zetanot,&SN,&CN,&DN);
        if ( ABS(cos(PHI) - cos(TEST)) > 1e-2  ) S = -1*S;
        printf("S: %.14e \n",S);
        while( Yn <= Y)
        {
           i = solve(bhat,phi,s,S,&PHI,&TEST,yhat,b0hat,phi0,
                     PHI0,Gamma,mu0,h,khat,thetahat,zeta,zetanot,
                     &SN,&CN,&DN);
           phi[1] = phi[1] - h*mu[1]*zeta/cg[1];
           phi[2] = phi[2] - h*mu[2]*zeta/cg[2];
           phi[3] = phi[3] - h*mu[3]*zeta/cg[3];
           i = unscale(b,bhat,K0);
           i = output(Y,h,b,phi,PHI,TEST,SN,CN,DN);
           Y = Y + step_size;
           zeta  = Y - Ys;
        }
        Yp = Yn;
        b0hat[1] = bhat[1];
        b0hat[2] = bhat[2];
        b0hat[3] = bhat[3];
        phi0[1] = phi[1];
        phi0[2] = phi[2];
        phi0[3] = phi[3];
        PHI0 = PHI;
    }
fclose(in);
fclose(out);
}

error(int i)
{
printf("-ERROR- %3d \n",i);
exit(1);
}
```

```
/*************************************************************/
/* Name: init                                                */
/*                                                           */
/* Purpose: This function reads the input file and           */
/*          initializes the parameters and initial           */
/*          conditions.                                      */
/*                                                           */
/* Written: Feb 9, 1992                                      */
/* By: Francois Primeau                                      */
/*                                                           */
/*************************************************************/
#include <stdio.h>
#include <math.h>

init(FILE *in,
     double b0hat[4],
     double w[4],
     double K0[4],
     double *mu0,
     double mu[4],
     double cg[4],
     double *b,
     double *H1,
     double *H2,
     double k[4],
     double omega[4],
     double z[4]
     )
{
   int i;
   double temp;
   double N[4],c[4];

   /*read in the first wave number*/
   fscanf(in,"%le ",&k[1]);

   /*read in the parameters for the exponential
     cross shelf profile*/
   fscanf(in,"%le %le ",b,H2);
   *H1 = *H2/exp(2*(*b));

   /*compute the base triad*/
   i = triad(k,omega,z,*b);

   swap(&k[2],&k[3]);
   swap(&k[1],&k[2]);
   swap(&omega[2],&omega[3]);
```

```
    swap(&omega[1],&omega[2]);
    swap(&z[2],&z[3]);
    swap(&z[1],&z[2]);


    /*read in the initial wave amplitude magnitudes*/
    fscanf(in,"%le %le %le",&b0hat[1],&b0hat[2],&b0hat[3]);

    /*read in the initial wave amplitude phases*/
    fscanf(in,"%le %le %le",&w[1],&w[2],&w[3]);

    /*calculate the interaction coefficients*/
    i = K0j(K0,*b,*H2,k,omega,z);

    /*calculate alongshore topography coefficient*/

    i = Nj(N,*b,*H2,z);
    i = cj(c,k,omega);
    i = cgj(cg,c,*H2,N,k,z);
    i = mu0j(mu,*b,*H2,N,k,omega,z);

    *mu0 = mu[1]/cg[1] + mu[2]/cg[2] + mu[3]/cg[3];


    return(0);
}
```

```
/*************************************************************/
/* Name: scale                                               */
/*                                                           */
/* Purpose: This function scales the wave amplitudes.        */
/*                                                           */
/* Written: Feb 19, 1992                                     */
/* By: Francois Primeau                                      */
/*                                                           */
/*************************************************************/
#include <stdio.h>
#include <math.h>

int scale(double bhat[4],
          const double b[4],
          const double K0[4]
         )
{
   bhat[1] = b[1]*sqrt(_ABS(K0[2]*K0[3]));
   bhat[2] = b[2]*sqrt(_ABS(K0[3]*K0[1]));
   bhat[3] = b[3]*sqrt(_ABS(K0[1]*K0[2]));

   return(0);
}
```

```
/*****************************************************************/
/* Name: astopo                                                  */
/*                                                               */
/* Purpose: This function reads in the along shore topography    */
/*          data.                                                */
/*                                                               */
/* Written: Feb 19, 1992                                         */
/* By: Francois Primeau                                          */
/*                                                               */
/*****************************************************************/
#include <stdio.h>
#include <math.h>
int astopo(FILE *in,
           double *Yn,
           double *Ys,
           double *zetanot,
           double *H,
           double *step
          )
{
  if (fscanf(in,"%le %le %le %le %le ",Yn,Ys,zetanot,H,step)
      != 5)
      return(1);
  return(0);
}
```

```
/*****************************************************************/
/* Name: cj                                                      */
/*                                                               */
/* Purpose: This function calculates the phase speeds.           */
/*          The function returns the value 0 in all cases.*/
/*                                                               */
/* Written: Feb 9, 1992                                          */
/* By: Francois Primeau                                          */
/*                                                               */
/*****************************************************************/
#include <stdio.h>
#include <math.h>
int cj( double c[4],              /*phase speeds */
        const double k[4],        /*wave numbers*/
        const double omega[4] /*frequencies*/
      )
{
   c[1] = omega[1]/k[1];
   c[2] = omega[2]/k[2];
   c[3] = omega[3]/k[3];

   printf("c1 %e \n",c[1]);
   printf("c2 %e \n",c[2]);
   printf("c3 %e \n",c[3]);

   return(0);
}
```

```
/*****************************************************************/
/* Name: zj                                                      */
/*                                                               */
/* Purpose: This function calculates the roots of the            */
/*          transcendental equation                              */
/*                      tan(zj) = - zj/(b + |kj|).               */
/*          zj>0, given the dispersion relation.                 */
/*          The function returns the value 0 in all cases.      */
/*                                                               */
/* Written: Feb 9, 1992                                          */
/* By: Francois Primeau                                          */
/*                                                               */
/*****************************************************************/
#include <stdio.h>
#include <math.h>
int zj( double z[4],          /*roots*/
        const double b,    /*cross shelf profile parameter*/
        const double k[4],   /*wave numbers */
        const double omega[4]/*frequency*/
      )
{
   z[1] = sqrt(-2*b*k[1]/omega[1]-k[1]*k[1]-b*b);/*disp. rel.
                                                   for wave 1*/
   z[2] = sqrt(-2*b*k[2]/omega[2]-k[2]*k[2]-b*b);/*disp. rel.
                                                   for wave 2*/
   z[3] = sqrt(-2*b*k[3]/omega[3]-k[3]*k[3]-b*b);/*disp. rel.
                                                   for wave 3*/

   printf("z1 = %e \n",z[1]);
   printf("z2 = %e \n",z[2]);
   printf("z3 = %e \n",z[3]);

   return(0);
}
```

```
/*****************************************************************/
/* Name: Nj                                                      */
/*                                                               */
/* Purpose: This function calculates the normalization           */
/*          constants as given by Hsieh and Mysak (JPO 1980).*/
/*          The function returns the value 1 in all cases.       */
/*                                                               */
/* Written: Feb 9, 1992                                          */
/* By: Francois Primeau                                          */
/*                                                               */
/*****************************************************************/
#include <stdio.h>
#include <math.h>
int Nj( double N[4],          /*Normalization constants*/
        const double b,       /*cross shelf profile parameter*/
        const double H2,      /*cross shelf profile parameter*/
        const double z[4]     /*roots*/
      )
{

   N[1] = sqrt(2*z[1]*H2/(b*(2*z[1]-sin(2*z[1]))));
   N[2] = sqrt(2*z[2]*H2/(b*(2*z[2]-sin(2*z[2]))));
   N[3] = sqrt(2*z[3]*H2/(b*(2*z[3]-sin(2*z[3]))));

   printf("N1 %e \n",N[1]);
   printf("N2 %e \n",N[2]);
   printf("N3 %e \n",N[3]);

   return(0);
}
```

```
/*****************************************************************/
/* Name: cgj                                                     */
/*                                                               */
/* Purpose: This function calculates the groups velocities       */
/*          using the formulas given by Hsieh and Mysak          */
/*          (JPO 1980).                                          */
/*                                                               */
/* Written: Feb 9, 1992                                          */
/* By: Francois Primeau                                          */
/*                                                               */
/*****************************************************************/
#include <stdio.h>
#include <math.h>
int cgj( double cg[4],        /*group velocites*/
         const double c[4],   /*phase speeds*/
         const double H2,     /*cross shelf profile parameter*/
         const double N[4],   /*normalization constants*/
         const double k[4],   /*wave numbers*/
         const double z[4]    /*roots*/
       )
{
   double gamma[4];

   gamma[1] = (N[1]*N[1]/H2)*(0.5-sin(2*z[1])/(4*z[1]) +
                              sin(z[1])*sin(z[1])/(2*_ABS(k[1])));
   gamma[2] = (N[2]*N[2]/H2)*(0.5-sin(2*z[2])/(4*z[2]) +
                              sin(z[2])*sin(z[2])/(2*_ABS(k[2])));
   gamma[3] = (N[3]*N[3]/H2)*(0.5-sin(2*z[3])/(4*z[3]) +
                              sin(z[3])*sin(z[3])/(2*_ABS(k[3])));

   cg[1] = c[1]*(1+2*gamma[1]*c[1]*k[1]*k[1]);
   cg[2] = c[2]*(1+2*gamma[2]*c[2]*k[2]*k[2]);
   cg[3] = c[3]*(1+2*gamma[3]*c[3]*k[3]*k[3]);

   printf("cg1 %e \n",cg[1]);
   printf("cg2 %e \n",cg[2]);
   printf("cg3 %e \n",cg[3]);

   return(0);
}
```

```
/*****************************************************************/
/* Name: mu0j                                                    */
/*                                                               */
/* Purpose: This function calculates the alongshore              */
/*          topogaraphy coefficients.                            */
/*          coeffs.                                              */
/*                                                               */
/* Written: Feb 9, 1992                                          */
/* By: Francois Primeau                                          */
/*                                                               */
/*****************************************************************/
#include <stdio.h>
#include <math.h>
int mu0j( double mu[4],     /*alongshore topography coeffs*/
          const double b,  /*cross shelf profile parameter*/
          const double H2, /*cross shelf profile parameter*/
          const double N[4],/*normalization constants*/
          const double k[4],/*wave numbers*/
          const double omega[4],/*frequencies*/
          const double z[4]     /*roots*/
        )
{
   int j;

   for(j = 1; j<= 3;j++)
      mu[j] = (N[j]*N[j]*omega[j]/(4*H2*H2))*(
              (b*b-k[j]*k[j]-z[j]*z[j])/b +
              exp(2*b)*(-b*b+k[j]*k[j]-z[j]*z[j])/b +
              ( b*exp(2*b)*(b*b-k[j]*k[j]-3*z[j]*z[j]) +
               b*(-b*b+k[j]*k[j]+3*z[j]*z[j])*cos(2*z[j]) +
               z[j]*(3*b*b-k[j]*k[j]-z[j]*z[j])*sin(2*z[j])
              )/(b*b+z[j]*z[j]));

   return(0);
}
```

```
/*****************************************************************/
/* Name: muj                                                     */
/*                                                               */
/* Purpose: This function calculates the alongshore topog        */
/*          topography coefficient.                              */
/*                                                               */
/* Written: Feb 9, 1992                                          */
/* By: Francois Primeau                                          */
/*                                                               */
/*****************************************************************/
#include <stdio.h>
#include <math.h>
int muj( double *mu0,        /*alongshore topography coeff*/
         const double b,     /*cross shelf profile parameter*/
         const double H2     /*cross shelf profile parameter*/
         const double k[4], /*wave numbers*/
         const double omega[4],/*frequencies*/
         const double z[4]  /*roots*/
       )
{
  int i;
  double c[4];  /*phase velocities*/
  double cg[4]; /*group velocities*/
  double N[4];  /*Normalization constants*/
  double mu[4]; /*alongshore topography coefficients*/


  i = Nj(N,b,H2,z);
  i = cj(c,k,omega);
  i = cgj(cg,c,H2,N,k,z);
  i = mu0j(mu,b,H2,N,k,omega,z);

  *mu0 = mu[1]/cg[1] + mu[2]/cg[2] + mu[3]/cg[3];

  return(0);
}
```

```
/*****************************************************************/
/* Name: K0j                                                     */
/*                                                               */
/* Purpose: This function calculates the interaction             */
/*          coefficients for the three wave interaction          */
/*          equations, using the formulas given by               */
/*          Hsieh and Mysak (JPO 1980). The function returns*/
/*          the value 0 in  all cases.                           */
/*                                                               */
/* Written: Feb 9, 1992                                          */
/* By: Francois Primeau                                          */
/*                                                               */
/*****************************************************************/
#include <stdio.h>
#include <math.h>
/*interaction coefficients*/
static double K[4][4][4] = {
          {{0,0,0,0},{0,0,0,0},{0,0,0,0},{0,0,0,0}},
          {{0,0,0,0},{0,0,0,0},{0,0,0,0},{0,0,0,0}},
          {{0,0,0,0},{0,0,0,0},{0,0,0,0},{0,0,0,0}},
          {{0,0,0,0},{0,0,0,0},{0,0,0,0},{0,0,0,0}}};
int K0j( double K0[4],      /*interaction coefficient*/
         const double b,    /*cross shelf profile parameter*/
         const double H2,   /*cross shelf profile parameter*/
         const double k[4], /*wave numbers*/
         const double omega[4],/*frequencies*/
         const double z[4]     /*roots*/
       )
{
  int i;
  double Ks[4]; /*Kj as defined by Hsieh and Mysak*/
  double c[4];  /*phase velocities*/
  double cg[4]; /*group velocities*/
  double N[4];  /*Normalization constants*/

  i = Nj(N,b,H2,z);
  i = cj(c,k,omega);
  i = cgj(cg,c,H2,N,k,z);
  i = Kjlm(K,b,H2,N,k,z);

  Ks[1] = c[1]*(K[1][2][3] + K[1][3][2]);
  Ks[2] = c[2]*(K[2][3][1] + K[2][1][3]);
  Ks[3] = c[3]*(K[3][1][2] + K[3][2][1]);

  K0[1] = Ks[1]/cg[1];
  K0[2] = Ks[2]/cg[2];
  K0[3] = Ks[3]/cg[3];
```

```
    printf("K1 = %e \n",Ks[1]);
    printf("K2 = %e \n",Ks[2]);
    printf("K3 = %e \n",Ks[3]);

    printf("K01 = %e \n",K0[1]);
    printf("K02 = %e \n",K0[2]);
    printf("K03 = %e \n",K0[3]);

    return(0);
}
```

```
/*****************************************************************/
/* Name: Kjlm                                                    */
/*                                                               */
/* Purpose: This function calculates the Kjlm coefficients       */
/*          using the formulas given by Hsieh and Mysak          */
/*          (JPO 1980). The function returns the value 0 in      */
/*           all cases.                                          */
/*                                                               */
/* Written: Feb 9, 1992                                          */
/* By: Francois Primeau                                          */
/*                                                               */
/*****************************************************************/
#include <stdio.h>
#include <math.h>
const int permute[6][3] = {{1,2,3},{1,3,2},{2,3,1},{2,1,3},
                           {3,1,2},{3,2,1}};
int Kjlm( double K[4][4][4],   /*interaction coefficients*/
          const double b,   /*cross shelf profile prmtr*/
          const double H2,  /*cross shelf profile prmtr*/
          const double N[4],/*normalization constants*/
          const double k[4],/*wave numbers*/
          const double z[4] /*roots*/
        )
{
   double b1,b2,b3,b4,A1,A2,A3,A4,B1,B2,B3,
          B4,C1,C2,C3,C4,I1,I2,I3;
   int i,j,l,m;

   for ( i = 0; i<6; i++) {
      j = permute[i][0];
      l = permute[i][1];
      m = permute[i][2];

      b1 = z[j]-z[l]-z[m];
      b2 = z[j]-z[l]+z[m];
      b3 = z[j]+z[l]-z[m];
      b4 = z[j]+z[l]+z[m];

      A1 = sin(b1)/(b*b+b1*b1);
      A2 = sin(b2)/(b*b+b2*b2);
      A3 = sin(b3)/(b*b+b3*b3);
      A4 = sin(b4)/(b*b+b4*b4);

      B1 = cos(b1)/(b*b+b1*b1);
      B2 = cos(b2)/(b*b+b2*b2);
      B3 = cos(b3)/(b*b+b3*b3);
      B4 = cos(b4)/(b*b+b4*b4);
```

```
        C1 = exp(b)/(b*b+b1*b1);
        C2 = exp(b)/(b*b+b2*b2);
        C3 = exp(b)/(b*b+b3*b3);
        C4 = exp(b)/(b*b+b4*b4);

        I1 = 0.25*((b*A1+b1*B1-b1*C1)-(b*A2+b2*B2-b2*C2)
                  -(b*A3+b3*B3-b3*C3)+(b*A4+b4*B4-b4*C4));
        I2 = 0.25*((b1*A1-b*B1+b*C1)-(b2*A2-b*B2+b*C2)
                  +(b3*A3-b*B3+b*C3)-(b4*A4-b*B4+b*C4));
        I3 = 0.25*(-(b1*A1-b*B1+b*C1)+(b2*A2-b*B2+b*C2)
                  +(b3*A3-b*B3+b*C3)-(b4*A4-b*B4+b*C4));

        K[j][l][m] = N[j]*N[l]*N[m]*(k[m]/(H2*H2))
          *(
            b*(-3*b*b-z[j]*z[j]-2*z[l]*z[l]-
            3*k[l]*k[l]-k[m]*k[m])*I1
            +z[l]*(-b*b-z[j]*z[j]+k[l]*k[l]-k[m]*k[m])*I2
            +2*z[j]*(-b*b-z[l]*z[l])*I3
            +sin(z[j])*sin(z[l])*sin(z[m])/
            (_ABS(k[j])+_ABS(k[l])+_ABS(k[m]))
            *(k[l]*k[i]*(-2*_ABS(k[j])-_ABS(k[l]))+
            _ABS(k[l])*(k[m]*k[m]-k[j]*k[j]))
          );
        printf("K%d%d%d = %e \n",j,l,m,K[j][l][m]);
    }
    return(0);
}
```

```
/*************************************************************/
/* Name: invsn                                               */
/*                                                           */
/* Purpose: This function is the inverse Jacobi sn(*|*)      */
/*          elliptic function.                               */
/*                                                           */
/* Written: Feb 9, 1992                                      */
/* By: Francois Primeau                                      */
/*                                                           */
/*************************************************************/
#include <stdio.h>
#include <math.h>
int invsn(double *F,
          const double u,
          const double k
                      )
{
  int ierr;
  double l,phi,psi;
  double EE,FF;
  if (u > 1 ) return(4);
  phi = asin(u);
  psi = (M_PI/2) - phi;
  if (k > 1 ) return(1);
  l = sqrt(1 - k*k);
  dellpi_(&phi,&psi,&k,&l,&FF,&EE,&ierr);
  *F = FF;
  printf("%.14e\n",*F);
  return(ierr);
}
```

```
    i = sort(y);
    if (y[2] > 0) {
      printf("WARNING: %le   yhat[2] set to zero.\n",
             y[2]); y[2] = 0;}
    return(0);
}
```

```
/*****************************************************************/
/* Name: cubic                                                   */
/*                                                               */
/* Purpose: This function calculates the roots of the cubic      */
/*          polynomial.                                          */
/*                                                               */
/* Written: Feb 9, 1992                                          */
/* By: Francois Primeau                                          */
/*                                                               */
/*****************************************************************/
#include <stdio.h>
#include <math.h>
extern struct global;
int cubic(double y[4], /*roots*/
          const double b10,/*initial condition for wave1*/
          const double b20,/*initial condition for wave2*/
          const double b30,/*initial condition for wave3*/
          const double Gamma,/*constant of the motion*/
          const double mu0,/*alongshore
                             topographic coefficient*/
          const double H /*height of the top hat topographic
                           configuration*/
       )
{
   double d1,d2,d3; /*coefficients of the cubic polynomial*/
   double Q,R,D,M,theta;
   int i;

   /*initialize the coefficient of y to the power 2*/
   d1 = -b10*b10 + b20*b20 + b30*b30 + mu0*mu0*H*H/4;

   /*initialize the coefficient of y to the power 1*/
   d2 = -b10*b10*b20*b20 - b10*b10*b30*b30 + b20*b20*b30*b30 +
        Gamma*mu0*H - b10*b10*mu0*mu0*H*H/2;
   /*initialize the coefficient of y to the power 0*/
   d3 = -b10*b10*b20*b20*b30*b30 + Gamma*Gamma -
         b10*b10*Gamma*H*mu0 +
         b10*b10*b10*b10*mu0*mu0*H*H/4;

   Q = (3.0*d2-d1*d1)/9.0;
   R = (9.0*d1*d2-27*d3-2.0*d1*d1*d1)/54.0;
   D = Q*Q*Q+R*R ;
   if ( D>=0) return(1); /*roots are not all real*/
   theta = acos(R/sqrt(-Q*Q*Q));
   y[1] = 2*sqrt(-Q)*cos(theta/3) - d1/3;
   y[2] = 2*sqrt(-Q)*cos(theta/3 + 2*M_PI/3) - d1/3;
   y[3] = 2*sqrt(-Q)*cos(theta/3 + 4*M_PI/3) - d1/3;
```

```
/*****************************************************************/
/* Name: sort                                                    */
/*                                                               */
/* Purpose: This function sorts three numbers in descending      */
/*          order.                                               */
/*                                                               */
/* Written: Feb 9, 1992                                          */
/* By: Francois Primeau                                          */
/*                                                               */
/*****************************************************************/
#include <stdio.h>
#include <math.h>
int sort(double y[4]
        )
{
   if (y[2] > y[1]) swap(&y[1],&y[2]);
   if (y[3] > y[2]) swap(&y[2],&y[3]);
   if (y[2] > y[1]) swap(&y[1],&y[2]);
   return(0);
}

int swap(double *a,
         double *b)
{
double temp;
temp = *a;
*a = *b;
*b = temp;
return(0);
}
```

```
/*****************************************************************/
/* Name: solve                                                   */
/*                                                               */
/* Purpose: This function computes the amplitudes and the        */
/*          phases for the   resonant triad of waves.            */
/*                                                               */
/* Written: Feb 19, 1992                                         */
/* By: Francois Primeau                                          */
/*                                                               */
/*****************************************************************/
#include <stdio.h>
#include <math.h>
static struct global { double root[4];
                       double b0hat[4];
                       double khat;
                       double s[4];
                       double S;
                       double thetahat;
                       double Gamma;
                       double mu0;
                       double H;
                       double PHI;
                       double SN;
                       double CN;
                       double DN;
                     } common;

int solve( double bhat[4],
           double phi[4],
           double s[4],
           const double S,
           double *PHI,
           double *TEST,
           const double root[4],
           const double b0hat[4],
           const double phi0[4],
           const double PHI0,
           const double Gamma,
           const double mu0,
           const double H,
           const double khat,
           const double thetahat,
           const double zeta,
           const double zetanot,
           double *SN,
           double *CN,
           double *DN
```

```
        )
{
  extern double f1();
  extern double f2();
  extern double f3();
  extern double f4();
  double aerr = 10e-12;
  double rerr = 10e-12;
  double z;
  double err;
  int num;
  int ierr;
  int l = 1000;
  int m = 4000;
  int n;
  int iwk[1000];
  double wk[4000];
  double aa,bb;
  double ee;
  double er;
  {
    int i;
    for (i=1; i<4; i++)
    {
       common.root[i] = root[i];
       common.b0hat[i] = b0hat[i];
       common.s[i] = s[i];
    }
    common.khat = khat;
    common.thetahat = thetahat;
    common.Gamma = Gamma;
    common.mu0 = mu0;
    common.H = H;
    common.S = S;
  }
  aa = zetanot;
  bb = zeta;

  dqags_(f1,&aa,&bb,&aerr,&rerr,&z,&err,&num,
         &ierr,&l,&m,&n,iwk,wk);
  if (ierr !=0 ) printf("ierr1: %d %.14e \n",ierr,bb);
  phi[1] = phi0[1] + z;

  dqags_(f2,&aa,&bb,&aerr,&rerr,&z,&err,&num,
         &ierr,&l,&m,&n,iwk,wk);
  if (ierr != 0) printf("ierr2: %d bb: %.14e  \n",ierr,bb);
  phi[2] = phi0[2] + z;
```

```
    dqags_(f3,&aa,&bb,&aerr,&rerr,&z,&err,&num,
           &ierr,&l,&m,&n,iwk,wk);
    if (ierr !=0) printf("ierr3: %d bb: %.14e \n",ierr,bb);
    phi[3] = phi0[3] + z;
    dqags_(f4,&aa,&bb,&aerr,&rerr,&z,&err,&num,
           &ierr,&l,&m,&n,iwk,wk);
    if (ierr != 0) printf("ierr4: %d bb: %.14e \n",ierr,bb);
    *PHI = PHI0 - common.mu0*common.H*(zeta-zetanot) + z;


    if ( b(bhat,zeta,zetanot) == 1 ) return(1);
    *TEST= acos((Gamma-0.5*mu0*H*bhat[1]*bhat[1])/
    (bhat[1]*bhat[2]*bhat[3]));
    *SN = common.SN;
    *CN = common.CN;
    *DN = common.DN;
    s[1] = common.s[1];
    s[2] = common.s[2];
    s[3] = common.s[3];
    return(0);
}
double f1(double *Y)
{ double bhat[4];
  double value;
  int i;
  if ( (i=b(bhat,*Y,0)) != 0 ) error(100+i);
  return(
    -(2*common.Gamma-common.mu0*common.H*bhat[1]*bhat[1])/
     (2*bhat[1]*bhat[1]));
}
double f2(double *Y)
{ double bhat[4];
  double value;
  int i;
  if ( (i=b(bhat,*Y,0)) !=0 ) error(200+i);
  return(
    (2*common.Gamma-common.mu0*common.H*bhat[1]*bhat[1])/
    (2*bhat[2]*bhat[2]));
}
double f3(double *Y)
{ double bhat[4];
  double value;
  int i;
  if ( (i=b(bhat,*Y,0)) != 0 ) error(300+i);
  return(
    (2*common.Gamma-common.mu0*common.H*bhat[1]*bhat[1])/
```

```
    (2*bhat[3]*bhat[3]));
}
double f4(double *Y)
{ double bhat[4];
  double value;
  int i;
  if ( (i=b(bhat,*Y,0)) != 0 ) error(400+i);
  return(
    (common.Gamma - common.mu0*common.H*bhat[1]*bhat[1]*0.5)*
    (-1/(bhat[1]*bhat[1]) +
        1/(bhat[2]*bhat[2]) + 1/(bhat[3]*bhat[3])));
}
int b( double bhat[4],
       const double zeta,
       const double zetanot
     )
{
  double yhat;
  double sn,cn,dn,s,c,d;
  double temp;
  double u,u1,u2,u3;
  double k;
  double l;
  double psi,cpsi;
  double FF,EE;
  double flip1 =1;
  double flip2 =1;
  double flip3 =1;
  int ierr;

  u = common.S*sqrt(common.root[1]-common.root[3])*
      (zeta-zetanot)+common.thetahat;
  k = common.khat;

  if (k > 1) return(1);
  l = sqrt(1 - k*k);
  psi = M_PI/2;
  cpsi = 0.0;
  dellpi_(&psi,&cpsi,&k,&l,&FF,&EE,&ierr);
  while ( u > 4*FF) u = u - 4*FF;
  while ( u < 0.0)u = u + 4*FF;
  u1 = u;
  flip1 =1;
  if (u1 > 2*FF ){
     u1 = u1 -2*FF;
     flip1 = -1*flip1;
  }
```

```
if ( u1 > FF ) u1 = -1*u1 + 2*FF;
u2 = u;
flip2 = 1;
if (u2 > 2*FF) u2 = -u2 + 4*FF;
if (u2 > FF ) {
   u2 = -1*u2 + 2*FF;
   flip2 = -1*flip2;
}
u3 = u;
flip3 =1;
if ( u3 > 2*FF) u3 = u3 -2*FF;
if ( u3 > FF) u3 = -1*u3 + 2*FF;

djcbyf_(&u1,&k,&s,&c,&d,&ierr);
if (ierr != 0) return(ierr);
sn = flip1*s;
djcbyf_(&u2,&k,&s,&c,&d,&ierr);
if (ierr != 0) return(ierr);
cn = flip2*c;
djcbyf_(&u3,&k,&s,&c,&d,&ierr);
if (ierr != 0) return(ierr);
dn = flip3*d;
common.SN = sn;
common.CN = cn;
common.DN = dn;
if ( (_ABS(common.root[1] -
          common.b0hat[1]*common.b0hat[1]) < 1e-13 )
    & (sn < 0.0 )) common.s[1] = -1.0;
if ( (_ABS(common.root[1] -
          common.b0hat[1]*common.b0hat[1]) < 1e-13 )
    & (sn > 0.0 )) common.s[1] = 1.0;
if ( (_ABS(common.root[1] -
          common.b0hat[1]*common.b0hat[1]) < 1e-13 )
    & (cn < 0.0 )) common.s[3] = -1.0;
if ( (_ABS(common.root[1] -
          common.b0hat[1]*common.b0hat[1]) < 1e-13 )
    & (cn > 0.0 )) common.s[3] = 1.0;
yhat = (common.root[2]-common.root[1])*sn*sn+common.root[1];

bhat[1] = common.s[1]*
          sqrt(common.b0hat[1]*common.b0hat[1]-yhat);
bhat[2] = common.s[2]*
          sqrt(common.b0hat[2]*common.b0hat[2]+yhat);
bhat[3] = common.s[3]*
          sqrt(common.b0hat[3]*common.b0hat[3]+yhat);

return(0);
}
```

```
/*************************************************************/
/* Name: unscale                                             */
/*                                                           */
/* Purpose: This function unscales the wave amplitudes.      */
/*                                                           */
/* Written: Feb 19, 1992                                     */
/* By: Francois Primeau                                      */
/*                                                           */
/*************************************************************/
#include <stdio.h>
#include <math.h>

int unscale(double b[4],
            const double bhat[4],
            const double K0[4]
           )
{
   b[1] = bhat[1]/sqrt(_ABS(K0[2]*K0[3]));
   b[2] = bhat[2]/sqrt(_ABS(K0[3]*K0[1]));
   b[3] = bhat[3]/sqrt(_ABS(K0[1]*K0[2]));

   return(0);
}
```

```
/*************************************************************/
/* Name: unscale                                             */
/*                                                           */
/* Purpose: This function unscales the wave amplitudes.      */
/*                                                           */
/* Written: Feb 19, 1992                                     */
/* By: Francois Primeau                                      */
/*                                                           */
/*************************************************************/
#include <stdio.h>
#include <math.h>

int unscale(double b[4],
            const double bhat[4],
            const double K0[4]
           )
{
   b[1] = bhat[1]/sqrt(_ABS(K0[2]*K0[3]));
   b[2] = bhat[2]/sqrt(_ABS(K0[3]*K0[1]));
   b[3] = bhat[3]/sqrt(_ABS(K0[1]*K0[2]));

   return(0);
}
```

```
/*****************************************************************/
/* Name: output                                                  */
/*                                                               */
/* Purpose: This function ouputs the results and plots the       */
/*          solutions.                                           */
/*                                                               */
/* Written: Feb 9, 1992                                          */
/* By: Francois Primeau                                          */
/*                                                               */
/*****************************************************************/
#include <stdio.h>
#include <math.h>
extern FILE *out;
int output(double YY,
           double h,
           double b[4],
           double phi[4],
           double PHI,
           double TEST,
           double SN,
           double CN,
           double DN
          )
{

fprintf(out,"%e %e %e %e %e %e %e %e %e %e\n",
             (float)YY,(float)h,(float)b[1],(float)b[2],
             (float)b[3],(float)phi[1],(float)phi[2],
             (float)phi[3],(float)PHI,(float)TEST);
return(0);
}
```